

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Чорноморський національний університет імені Петра Могили**  
**Факультет комп'ютерних наук**  
**Кафедра інженерії програмного забезпечення**

ДОПУЩЕНО ДО ЗАХИСТУ

Завідувач кафедри інженерії  
програмного забезпечення

\_\_\_\_\_ Євген ДАВИДЕНКО

«\_\_» \_\_\_\_\_ 2024 р.

**КВАЛІФІКАЦІЙНА РОБОТА**  
**НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ МАГІСТРА**  
**ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ІДЕНТИФІКАЦІЇ РОСЛИН НА ОСНОВІ**  
**МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ**

Спеціальність 121 Інженерія програмного забезпечення  
Освітня програма «Інженерія програмного забезпечення»

**Здобувачка**

\_\_\_\_\_

**Аліна ГАЙСЮК**

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Керівник роботи**

канд. техн. наук,

доцент

\_\_\_\_\_

**Євген ДАВИДЕНКО**

«\_\_» \_\_\_\_\_ 20\_\_ р.

**Миколаїв – 2024**

## **Завдання на виконання кваліфікаційної роботи**

Чорноморський національний університет імені Петра Могили

Факультет	Комп'ютерних наук
Кафедра	Інженерії програмного забезпечення
Рівень вищої освіти	Другий (магістерський)
Освітній ступінь	Магістр
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення

**ЗАТВЕРДЖУЮ**

Завідувач кафедри інженерії  
програмного забезпечення

\_\_\_\_\_ Євген ДАВИДЕНКО

«\_\_\_» \_\_\_\_\_ 2024 р.

### **ЗАВДАННЯ**

**на кваліфікаційну магістерську роботу здобувача вищої освіти**

**Гайсюк Аліни**

---

1. Тема кваліфікаційної роботи «Програмне забезпечення ідентифікації рослин на основі методів штучного інтелекту» затверджена наказом ректора ЧНУ ім. Петра Могили № 220 від «04» вересня 2024 р.

2. Строк представлення кваліфікаційної роботи «19» грудня 2024 р.

3. Очікуваний результат: розроблений застосунок, який дозволяє ідентифікувати рослини на основі фото за допомогою методів машинного навчання; підвищення ефективності та точності процесу визначення рослин; забезпечення доступності застосунку для широкого кола користувачів, зокрема дослідників, студентів, садівників та любителів природи.

Початкові дані роботи: розробка концепції мобільного застосунку для

ідентифікації рослин із використанням методів штучного інтелекту; аналіз сучасних технологій у сфері комп'ютерного зору та обробки зображень; інтеграція наукових даних про рослини та створення інструментів для зручного внесення нової інформації; забезпечення зручного та інтуїтивно зрозумілого інтерфейсу для користувачів.

4. Перелік питань, що підлягають розробці:

- аналіз сучасних підходів і методів штучного інтелекту для ідентифікації об'єктів;
- визначення функціональних і технічних вимог до застосунку;
- огляд і аналіз аналогів систем для ідентифікації рослин;
- розробка програмного забезпечення для збору, обробки та класифікації зображень рослин;
- впровадження інструментів для навчання та тестування моделей штучного інтелекту;
- тестування мобільного застосунку з реальними даними;
- аналіз результатів роботи системи та розробка рекомендацій для подальшого вдосконалення.

5. Перелік графічних матеріалів: презентація, діаграми та схеми, приклади інтерфейсу користувача.

6. Консультанти:

<b>Консультант</b>	<b>Кафедра (організація)</b>	<b>Частина роботи</b>

Дата видачі завдання « \_\_\_\_ » \_\_\_\_\_ 20\_\_ р.

## КАЛЕНДАРНИЙ ПЛАН

### виконання кваліфікаційної роботи

Тема: Програмне забезпечення ідентифікації рослин на основі методів штучного інтелекту.

№	Найменування роботи	Початок	Закінчення	Примітки
1.	Розробка та затвердження завдання на виконання КМР	25.06.2024 р.	30.06.2024 р.	Виконано
2.	Огляд літератури за темою роботи	13.09.2024 р.	19.09.2024 р.	Виконано
3.	Аналіз предметної області	24.09.2024 р.	26.09.2024 р.	Виконано
4.	Розробка проєктних рішень	28.09.2024 р.	30.09.2024 р.	Виконано
5.	Моделювання та конструювання ПЗ	01.10.2024 р.	06.10.2024 р.	Виконано
6.	Кодування, тестування розробленого ПЗ, аналіз результатів тестування, розробка керівництва користувача	11.10.2024 р.	07.11.2024 р.	Виконано
8.	Оформлення КМР та презентації	10.11.2024 р.	20.11.2024 р.	Виконано
9.	Попередній захист	28.11.2024 р.	28.11.2024 р.	Виконано
10.	Рецензування	05.12.2024 р.	09.12.2024 р.	Виконано
11.	Завершення оформлення КМР та презентації	09.12.2024 р.	14.12.2024 р.	Виконано
12.	Відгук керівника	16.12.2024 р.	18.12.2024 р.	Виконано
13.	Захист кваліфікаційної роботи	19.12.2024 р.	19.12.2024 р.	Виконано

Здобувачка \_\_\_\_\_

Аліна ГАЙСЮК

«\_\_» \_\_\_\_\_ 20\_\_ р.

Керівник роботи \_\_\_\_\_

Євген ДАВИДЕНКО

канд. техн. наук, \_\_\_\_\_

доцент

«\_\_» \_\_\_\_\_ 20\_\_ р.

## АНОТАЦІЯ

до кваліфікаційної магістерської роботи

«Програмне забезпечення ідентифікації рослин на основі методів штучного інтелекту»

Здобувачка 608 гр.: Гайсюк Аліна

Керівник: канд. техн. наук, доцент Давиденко Євген

Кваліфікаційна магістерська робота присвячена розробці програмного забезпечення для ідентифікації рослин на основі методів штучного інтелекту.

**Актуальність дослідження**, полягає у тому, що з кожним роком зростає інтерес до захисту довкілля, ботаніки та дослідження біорізноманіття. Однак, процес ідентифікації рослин залишається складним завданням, особливо для людей без фахової підготовки. Програмні забезпечення на основі штучного інтелекту дозволяють автоматизувати та прискорити цей процес, роблячи його доступним для широкої аудиторії — від науковців до любителів природи. Розробка таких застосунків, як “GreenEye”, є актуальною через потребу у швидкій та точній ідентифікації рослин у різних сферах: сільському господарстві, екології, освіті та наукових дослідженнях.

**Об’єктом дослідження** є процеси ідентифікації рослин на основі аналізу зображень.

**Предметом дослідження** є методи штучного інтелекту для класифікації та розпізнавання рослин.

**Мета роботи** полягає у підвищенні ефективності і точності розпізнавання рослин за рахунок створення системи, призначеної для ідентифікації рослин на основі методів штучного інтелекту за допомогою використання технологій апаратної та програмної реалізації React Native, NEST.JS та бази даних PostgreSQL.

Кваліфікаційна магістерська робота складається зі вступу, чотирьох розділів, висновків та переліку джерел посилання.

У вступі обґрунтовується актуальність теми, описуються поставлені задачі, предмет дослідження та об’єкт дослідження.

У першому розділі надається аналіз аналогічних застосунків для ідентифікації рослин, а також системи, що розробляється, визначаються вимоги до програмного забезпечення та встановлюються технічні обмеження для проєкту. Описуються вимоги до користувацького інтерфейсу та основні функції застосунку.

Другий розділ присвячено опису моделювання застосунку. Розглядаються етапи його розробки, надаються варіанти сценаріїв використання, створюються алгоритми для ідентифікації рослин та будуються відповідні діаграми взаємодії.

У третьому розділі описуються основні аспекти проєктування системи ідентифікації рослин. Розглядається архітектура системи, яка забезпечує її надійність, масштабованість і ефективність. Наводиться структура бази даних, яка дозволяє зберігати, обробляти та управляти даними про рослини, забезпечуючи швидкий доступ до інформації. Також детально описується процес розробки датасету, що включає збір, обробку та класифікацію даних, необхідних для навчання моделей штучного інтелекту.

Четвертий розділ присвячено інформації щодо програмної реалізації застосунку. Детально описуються програмні засоби, які використовувалися для розробки системи, акцентується їхня роль у забезпеченні функціональності та продуктивності. Крім того, розглядається процес програмної реалізації, зокрема розробка основних компонентів застосунку, їх інтеграція та впровадження системи.

У висновках проводиться аналіз роботи та отриманих результатів.

Кваліфікаційна робота викладена на 103 сторінках машинописного тексту, складається із вступу, 4 розділів, загальних висновків та переліку джерел посилання з 19 найменувань. Праця містить 23 таблиць та 46 рисунків.

Ключові слова: *програмне забезпечення ідентифікація рослин, штучний інтелект, нейронні мережі, машинне навчання, React Native, NEST.JS та PostgreSQL.*

## **ABSTRACT**

to the qualifying master's thesis

“Plant identification software based on artificial intelligence methods”

Applicant of group 608: Haisiuk Alina

Supervisor: Ph.D. tech. sciences, Associate Professor Davydenko Yevhen

The qualifying master's thesis is devoted to the development of software for plant identification based on artificial intelligence methods.

**The relevance** of the research lies in the fact that interest in environmental protection, botany and biodiversity research is growing every year. However, the process of plant identification remains a difficult task, especially for people without professional training. Artificial intelligence-based software allows you to automate and accelerate this process, making it accessible to a wide audience, from scientists to nature lovers. The development of such applications as "GreenEye" is relevant due to the need for quick and accurate identification of plants in various fields: agriculture, ecology, education and scientific research.

**The object** of research is plant identification processes based on image analysis.

**The subject** of research is artificial intelligence methods for plant classification and recognition.

**The purpose** of the work is to increase the efficiency and accuracy of plant recognition by creating a system designed for plant identification based on artificial intelligence methods using the hardware and software implementation technologies of React Native, NEST.JS and the PostgreSQL database.

The qualification work consists of an introduction, four sections, conclusions and a list of references.

The introduction substantiates the relevance of the topic, describes the tasks, the subject of research and the object of research.

The first section provides an analysis of similar applications for plant identification, as well as the system under development, defines the software requirements, and

establishes technical constraints for the project. The requirements for the user interface and the main functions of the application are described.

The second section is devoted to the description of application modeling. The stages of its development are considered, options for use scenarios are provided, algorithms for plant identification are created, and appropriate interaction diagrams are constructed.

The third chapter describes the main aspects of designing a plant identification system. The architecture of the system, which ensures its reliability, scalability and efficiency, is considered. A database structure is provided that allows storing, processing and managing plant data, providing quick access to information. The dataset development process is also described in detail, including the collection, processing and classification of data required for training artificial intelligence models.

The fourth section is devoted to information about the software implementation of the application. The software tools that were used to develop the system are described in detail, and their role in ensuring functionality and productivity is emphasized. In addition, the process of software implementation is considered, in particular, the development of the main components of the application, their integration and implementation of the system.

In the conclusions, an analysis of the work and the obtained results is carried out.

The qualification work is presented on 103 pages of typewritten text, consists of an introduction, 4 sections, general conclusions, a list of references with 19 titles. The work contains 23 tables and 46 figures.

*Keywords: plant identification software, artificial intelligence, neural networks, machine learning, React Native, NEST.JS, and PostgreSQL.*



## ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	3
ВСТУП.....	4
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ .....	7
1.1 Огляд та аналіз застосунків-аналогів .....	7
1.2 Аналіз системи, що розробляється .....	13
1.3 Специфікація вимог застосунка “GreenEye” .....	17
1.4 Аналіз методів штучного інтелекту для системи.....	28
Висновки до розділу 1 .....	30
2 МОДЕЛЮВАННЯ СИСТЕМИ ІДЕНТИФІКАЦІЇ РОСЛИН .....	31
2.1 Етапи реалізації проєкту .....	31
2.2 Створення USE CASE .....	32
2.3 Алгоритм роботи програмного застосунка.....	43
2.4 Розробка діаграми розгортання.....	47
2.5 Побудова та використання діаграм послідовності.....	51
Висновки до розділу 2 .....	55
3 ПРОЄКТУВАННЯ СИСТЕМИ ІДЕНТИФІКАЦІЇ РОСЛИН.....	56
3.1 Архітектура системи .....	56
3.2 Структура бази даних.....	57
3.3 Розробка датасету .....	67
Висновки до розділу 3 .....	70
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКА.....	71
4.1 Програмні засоби для розробки системи .....	71
4.2 Опис програмної реалізації .....	75
Висновки до розділу 4 .....	92
ВИСНОВКИ.....	93
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	94
ДОДАТОК А Апробація результатів КМР .....	96

**ПЕРЕЛІК СКОРОЧЕНЬ**

ПЗ	–	програмне забезпечення
БД	–	база даних
ШІ	–	штучний інтелект
ТЗ	–	технічне завдання
API	–	Application Programming Interface
AI	–	Artificial intelligence
CNN	–	Convolutional Neural Networks
OTP	–	One-Time Password

## ВСТУП

У сучасному світі технології штучного інтелекту та мобільних застосунків суттєво впливають на різні сфери життя, зокрема на ботаніку і природничі науки. Ідентифікація рослин є ключовою задачею для багатьох галузей, таких як сільське господарство, екологія, біологія та садівництво. Завдяки використанню новітніх мобільних технологій та алгоритмів машинного навчання, ідентифікація рослин стала значно доступнішою для широкого кола користувачів.

Актуальність цієї проблеми зростає з підвищенням інтересу до екологічної свідомості та бажанням людей краще розуміти природу. Проте, процес ідентифікації рослин часто вимагає спеціальних знань та навичок, які не завжди є у звичайних користувачів. Вирішенням цієї проблеми може стати створення мобільного застосунку, який допоможе в ідентифікації рослин за допомогою штучного інтелекту, аналізуючи зображення.

Застосунок “GreenEye” призначений для автоматизації процесу ідентифікації рослин. Він дозволяє користувачам робити фотографії рослин, а потім, за допомогою нейронних мереж, отримувати інформацію про їх вид. Мета розробки полягає у створенні зручного та ефективного інструменту для дослідження та вивчення рослин.

**Актуальність.** З кожним роком зростає інтерес до захисту довкілля, ботаніки та дослідження біорізноманіття. Однак, процес ідентифікації рослин залишається складним завданням, особливо для людей без фахової підготовки. Програмні забезпечення на основі штучного інтелекту дозволяють автоматизувати та прискорити цей процес, роблячи його доступним для широкої аудиторії — від науковців до любителів природи. Розробка таких застосунків, як “GreenEye”, є актуальною через потребу у швидкій та точній ідентифікації рослин у різних сферах: сільському господарстві, екології, освіті та наукових дослідженнях.

**Об’єктом дослідження** є процеси ідентифікації рослин на основі аналізу зображень.

**Предметом дослідження** є методи штучного інтелекту для класифікації та розпізнавання рослин.

**Мета роботи** полягає у підвищенні ефективності і точності розпізнавання рослин за рахунок створення системи, призначеної для ідентифікації рослин на основі методів штучного інтелекту за допомогою використання технологій апаратної та програмної реалізації React Native, NEST.JS та бази даних PostgreSQL.

Для досягнення цієї мети, у роботі поставлено та розв'язано наступні **завдання**:

- 1) провести аналіз існуючих рішень та аналогічних мобільних застосунків;
- 2) розробити технічне завдання на створення застосунку “GreenEye”;
- 3) розробити дизайн та інтерфейс користувацької частини застосунка;
- 4) реалізувати функціональні можливості застосунку, включаючи модуль ідентифікації рослин;
- 5) створити базу даних для збереження інформації про рослини;
- 6) забезпечити механізм автоматичної обробки зображень рослин за допомогою нейронних мереж;
- 7) реалізувати функції пошуку та надання додаткової інформації про ідентифіковані рослини;
- 8) протестувати застосунок та забезпечити його стабільну роботу.

#### **Сфера застосування:**

1) *освіта*: застосунок може бути корисним для шкіл, університетів, ботанічних садів та природничих музеїв. Він дозволяє учням і студентам легко ідентифікувати рослини під час практичних занять або польових досліджень, сприяючи покращенню процесу навчання біології та ботаніки;

2) *екологія та природоохоронні дослідження*: застосунок “GreenEye” може допомогти екологам, біологам та ботанікам швидко ідентифікувати рослинні види у польових умовах, що особливо важливо при дослідженні біорізноманіття,

моніторингу змін в екосистемах або під час роботи над проектами з охорони природи;

3) *сільське господарство*: фермерам і агрономам застосунок допомагає визначати рослини на полях, розпізнавати бур'яни та шкідливі види рослин. Це сприяє більш ефективному управлінню посівами та покращенню врожайності, дозволяючи швидше реагувати на зміни в агроландшафті;

4) *ландшафтний дизайн та садівництво*: для ландшафтних дизайнерів і садівників “GreenEye” стане корисним інструментом для ідентифікації рослин під час роботи на об'єктах. Він дозволяє точно визначати види рослин, що сприяє кращому плануванню озеленення територій;

5) *туризм та активний відпочинок*: любителі природи, туристи та мандрівники можуть використовувати застосунок для ідентифікації рослин під час прогулянок у лісах, горах та парках. Це допомагає краще розуміти природу навколо та підвищує інтерес до дослідження флори.

Таким чином, застосунок “GreenEye” стане значущим ресурсом для різноманітних груп користувачів, які прагнуть швидко і точно ідентифікувати рослини та отримати детальну інформацію про них. Застосунок надає можливість швидкого доступу до актуальної інформації про види рослин, їх характеристики та середовище проживання, що сприятиме підвищенню екологічної свідомості і підтримці наукових досліджень.

**Апробація результатів** КМР відбулась під час XXVII Всеукраїнської науково-практичної конференції «Могилянські читання – 2024», Миколаїв, 06–10 листопада, 2024 р. (Додаток А).

## 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

### 1.1 Огляд та аналіз застосунків-аналогів

Застосунки-аналоги для ідентифікації рослин є різноманітними платформами, що використовують технології комп'ютерного зору та штучного інтелекту для розпізнавання рослин на основі їх зображень. Ці системи мають різні функції та можливості, але їх спільна мета — забезпечити точність, зручність та доступність у процесі ідентифікації рослин. Аналіз та огляд таких застосунків можуть бути корисними для розробки нового застосунку, оскільки допомагають виявити ключові функції, можливі проблеми та області для вдосконалення.

Для огляду та аналізу були відібрані наступні популярні застосунки: PlantNet (табл. 1.1), iNaturalist (табл. 1.2), PictureThis (табл. 1.3).

#### **PlantNet**

*“PlantNet”* – один з найбільш відомих застосунків для ідентифікації рослин, який використовує технології комп'ютерного зору для визначення видів на основі зображень. Основні характеристики застосунку:

- 1) *база даних*: застосунок має велику базу даних зображень рослин, що постійно розширюється завдяки внескам користувачів [1];
- 2) *функції ідентифікації*: застосунок дозволяє завантажувати зображення рослин для їх автоматичного розпізнавання, надаючи інформацію про види, їхні характеристики та середовище проживання;
- 3) *інтерфейс користувача*: інтерфейс є простим і зрозумілим, що дозволяє легко завантажувати зображення та отримувати результати;
- 4) *соціальні функції*: користувачі можуть обмінюватися інформацією, коментувати спостереження та брати участь у спільнотах природолюбів (рис. 1.1).

Таблиця 1.1 – Основні характеристики застосунку “PlantNet”

<b>Назва</b>	PlantNet
<b>Розробник</b>	CIRAD (Centre de coopération Internationale en Recherche Agronomique pour le Développement), разом із іншими науковими установами.
<b>Країна розробника</b>	Франція
<b>Мова реалізації</b>	Python
<b>Архітектура</b>	Mobile application
<b>Сайт</b>	<a href="https://plantnet.org/en/">https://plantnet.org/en/</a>
<b>Перелік функцій, характеристик</b>	<ol style="list-style-type: none"> <li>1) Розпізнавання рослин за зображенням з наданням детальної інформації про види.</li> <li>2) Доступ до великої бази даних зображень та інформації про рослини.</li> <li>3) Можливість участі в спільнотах і обміну інформацією з іншими користувачами.</li> <li>4) Інтуїтивно зрозумілий інтерфейс для легкого завантаження зображень і отримання результатів.</li> </ol>
<b>Переваги</b>	<ol style="list-style-type: none"> <li>1) Велика база даних зображень рослин.</li> <li>2) Безкоштовний доступ та можливість участі в спільнотах.</li> <li>3) Простий і зручний інтерфейс.</li> </ol>
<b>Недоліки</b>	<ol style="list-style-type: none"> <li>1) Обмежене охоплення деяких рідкісних видів рослин.</li> <li>2) Може бути недостатньо точним у складних випадках.</li> </ol>

Використовуючи методи комп'ютерного зору та машинного навчання, програма може аналізувати зображення рослин і порівнювати їх із базою даних, що складається з тисяч видів рослин. Крім того, PlantNet дозволяє користувачам додавати свої спостереження та обмінюватися ними з іншими учасниками спільноти, що допомагає вдосконалювати і розширювати базу даних програми.

Застосунок “PlantNet” є одним із популярних інструментів для ідентифікації рослин, поєднуючи технології комп'ютерного зору та участь користувачів для постійного розширення бази даних. Завдяки простому інтерфейсу, багатій функціональності та можливості спільного обміну інформацією, він є зручним та доступним для широкого кола користувачів.

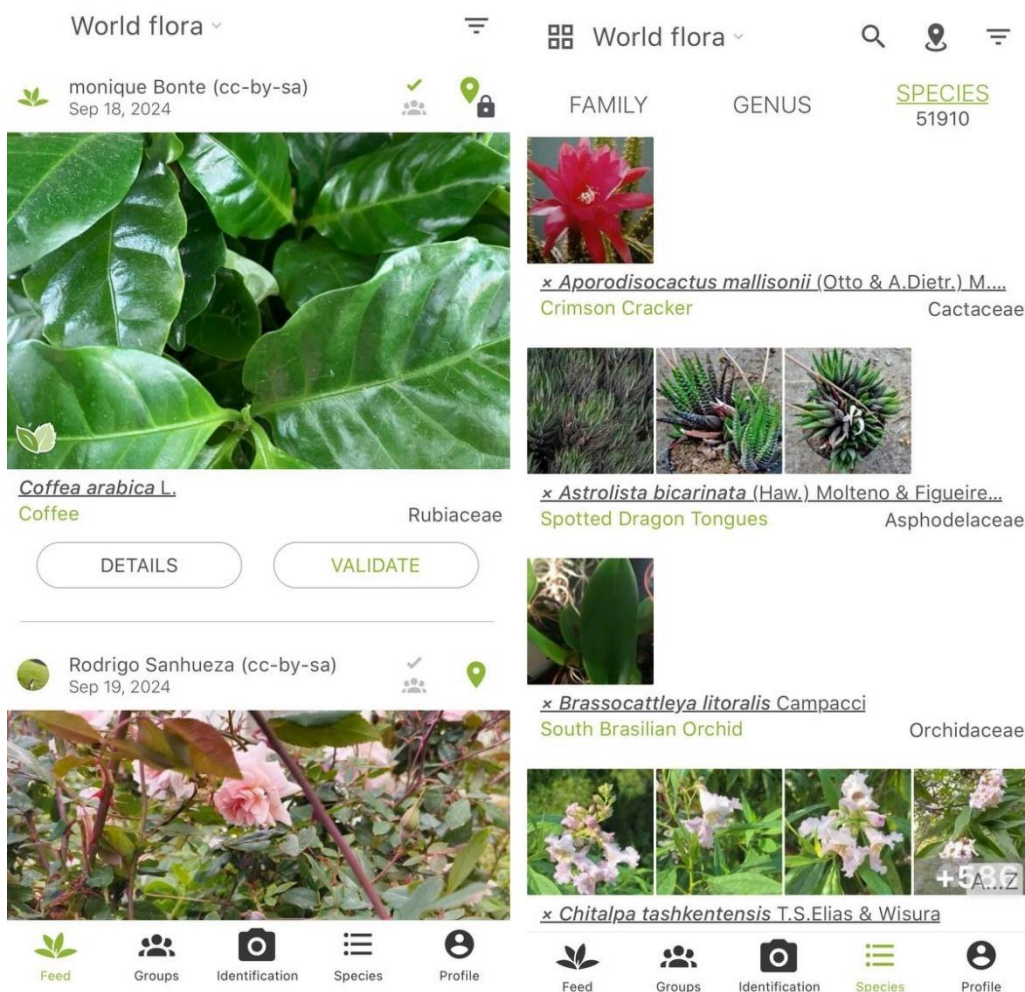


Рисунок 1.1 – Інтерфейс застосунку “PlantNet”

Незважаючи на певні обмеження в охопленні рідкісних видів, застосунок залишається провідним рішенням у галузі ботанічних досліджень та екологічної освіти.

### iNaturalist

“iNaturalist” – це платформа для вивчення природи, яка дозволяє користувачам завантажувати зображення рослин і тварин для їх ідентифікації, використовуючи спільноту експертів і технології штучного інтелекту. Основні характеристики:

- 1) *взаємодія з експертами*: застосунок поєднує автоматичне розпізнавання з можливістю отримання підтвердження від експертів [2];
- 2) *соціальні функції*: користувачі можуть обговорювати спостереження, отримувати рекомендації та брати участь у дослідницьких проєктах (рис. 1.2);



3) *навчальні ресурси*: застосунок пропонує освітні матеріали та допомагає користувачам дізнаватися більше про біологічні види, які вони зустрічають, сприяючи екологічній просвіті.

Таблиця 1.2 – Основні характеристики застосунку “iNaturalist”

<b>Назва</b>	iNaturalist
<b>Розробник</b>	California Academy of Sciences i National Geographic Society
<b>Країна розробника</b>	США
<b>Мова реалізації</b>	Ruby on Rails, JavaScript
<b>Архітектура</b>	Mobile application
<b>Сайт</b>	inaturalist.org
<b>Перелік функцій, характеристик</b>	<ol style="list-style-type: none"> <li>1) Розпізнавання різних видів рослин з можливістю отримання підтвердження від спільноти.</li> <li>2) Мобільний застосунок з інтегрованими соціальними функціями.</li> <li>3) Можливість участі в наукових проєктах та дослідженнях.</li> </ol>
<b>Переваги</b>	<ol style="list-style-type: none"> <li>1) Спільнота експертів для перевірки ідентифікацій.</li> <li>2) Наявність функцій для соціальної взаємодії та участі в дослідженнях.</li> </ol>
<b>Недоліки</b>	<ol style="list-style-type: none"> <li>1) Може бути залежним від активності спільноти для підтвердження ідентифікацій.</li> <li>2) Може мати проблеми з точністю у випадку рідкісних видів.</li> </ol>

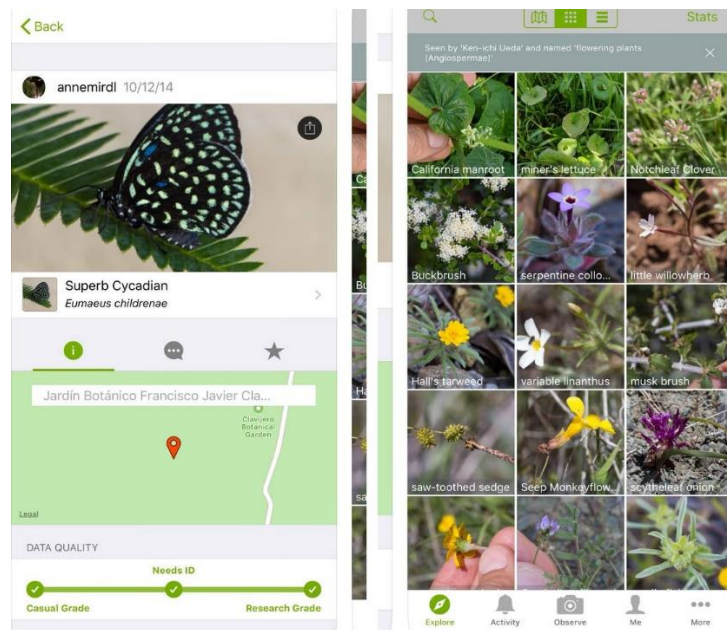


Рисунок 1.2 – Інтерфейс застосунку “iNaturalist”

Застосунок “iNaturalist” є гарною платформою, яка поєднує технології штучного інтелекту та взаємодію зі спільнотою експертів для ідентифікації рослин і тварин. Його головною перевагою є можливість отримання точного підтвердження від фахівців, що підвищує надійність результатів. Завдяки мобільній версії та соціальним функціям, “iNaturalist” сприяє активній участі користувачів у природничих дослідженнях, хоча залежність від активності спільноти може створювати певні обмеження у випадку підтвердження рідкісних видів.

### PictureThis

“PictureThis” – застосунок, що використовує штучний інтелект для розпізнавання рослин, надаючи користувачам інформацію про види та рекомендації щодо догляду. Основні характеристики:

- 1) *алгоритми штучного інтелекту*: використовуються для точного розпізнавання рослин та надання інформації [3];
- 2) *база даних*: наявність великої бази даних з описами рослин і порадами з догляду;
- 3) *інтерактивний інтерфейс*: інтуїтивно зрозумілий дизайн, що дозволяє легко фотографувати рослини, переглядати результати та отримувати потрібну інформацію за кілька секунд.

Таблиця 1.3 – Основні характеристики застосунку “PictureThis”

<b>Назва</b>	PictureThis
<b>Розробник</b>	Glority
<b>Країна розробника</b>	Китай
<b>Мова реалізації</b>	Java, Kotlin
<b>Архітектура</b>	Mobile application
<b>Сайт</b>	picturethisai.com
<b>Перелік функцій, характеристик</b>	1) Точне розпізнавання рослин завдяки алгоритмам штучного інтелекту. 2) Інформація про види, включаючи поради з догляду. 3) Мобільний застосунок з простим інтерфейсом.

## Кінець таблиці 1.3

<b>Переваги</b>	<ol style="list-style-type: none"> <li>1) Висока точність розпізнавання завдяки сучасним алгоритмам.</li> <li>2) Детальна інформація про догляд за рослинами.</li> <li>3) Простий та зручний інтерфейс.</li> </ol>
<b>Недоліки</b>	<ol style="list-style-type: none"> <li>1) Може вимагати підписки для доступу до всіх функцій.</li> <li>2) Деколи може бути менш точним у випадках з рідкісними видами.</li> </ol>

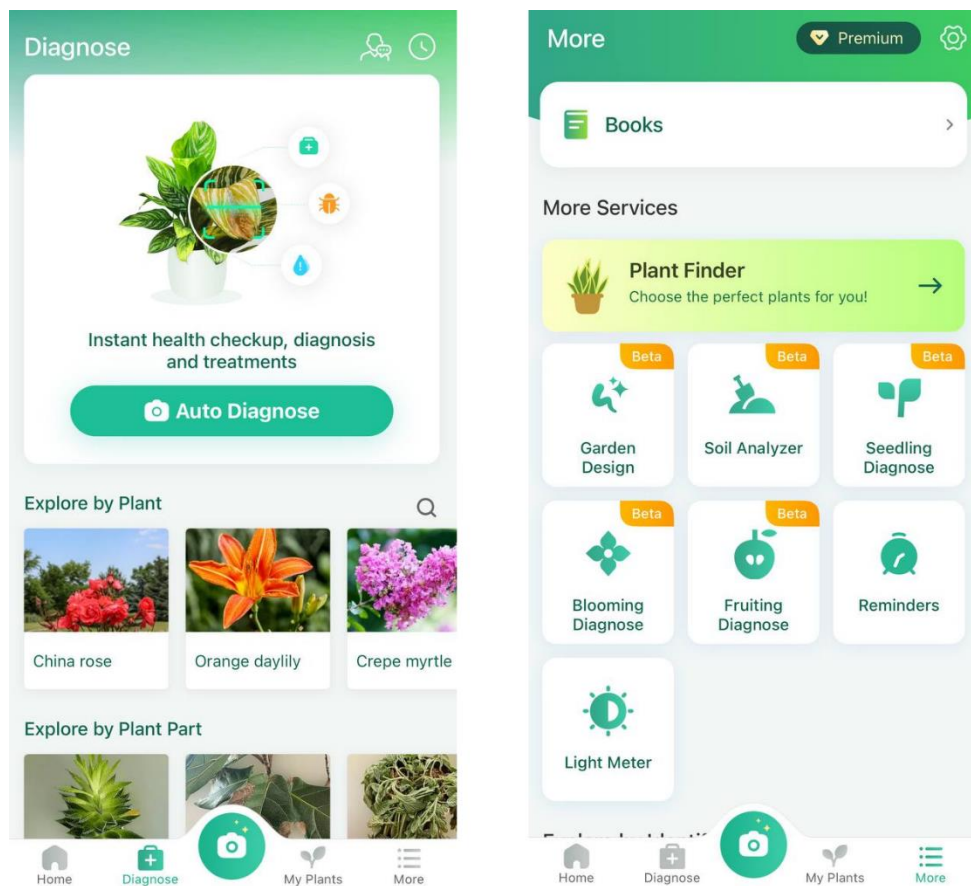


Рисунок 1.3 – Інтерфейс застосунку “PictureThis”

Застосунок “PictureThis” пропонує користувачам зручний інструмент для розпізнавання рослин і отримання рекомендацій щодо догляду за ними. Завдяки простому інтерфейсу, застосунок підходить як для новачків, так і для досвідчених користувачів. Однак наявність підписки для доступу до всіх функцій та можливі труднощі з розпізнаванням рідкісних видів можуть створювати певні обмеження.

## 1.2 Аналіз системи, що розробляється

Розробка системи “GreenEye” (рис. 1.4–1.5) спрямована на автоматизацію процесу ідентифікації рослин з використанням технології штучного інтелекту. Ця система орієнтована на використання на телефонах і планшетах, що забезпечить користувачам зручність під час польових досліджень або звичайних прогулянок на природі. “GreenEye” дозволяє ідентифікувати рослини за допомогою фото, що аналізуються нейронними мережами, надаючи користувачам точну інформацію про види рослин.

Застосунок розробляється на основі React Native, що забезпечує кросплатформену підтримку для операційних систем Android та iOS [4].

### Функціональні вимоги

1) *Підтримка мобільних пристроїв.* Застосунок розробляється для телефонів та планшетів, що використовують Android та iOS.

2) *Інтернет-з'єднання.* Для коректної роботи системи потрібно стабільне інтернет-з'єднання для завантаження зображень та отримання результатів ідентифікації.

3) *Інтерактивний інтерфейс.* Інтуїтивно зрозумілий дизайн, що дозволяє легко фотографувати рослини, переглядати результати та отримувати потрібну інформацію за кілька секунд.

Застосунок “GreenEye” передбачає наявність декількох ролей користувачів, кожна з яких має свої функції та права доступу. Це забезпечує гнучкість у використанні системи та можливість для різних категорій користувачів взаємодіяти з нею ефективно.

В системі присутні три ролі: адміністратор, основний користувач та науковий співробітник/фахівець.

**Адміністратор** відповідає за підтримку та управління всією системою. Його головні обов'язки включають оновлення бази даних рослин, налаштування нейронної мережі та забезпечення її безперебійної роботи, а також керування

обліковими записами користувачів, надання їм відповідних прав доступу та слідування за загальним станом системи.

*Функції:* додавання нових видів рослин до бази даних, оновлення та навчання нейронної мережі для підвищення точності ідентифікації, управління користувачами, зокрема видалення та редагування облікових записів, відстеження активності в системі та підтримка її роботи.

**Основні користувачі** – це люди, які використовують застосунок для ідентифікації рослин. Вони можуть робити фото рослин, надсилати їх для аналізу та отримувати детальну інформацію про види. Користувачі можуть зберігати свої результати для подальшого перегляду або порівняння.

*Функції:* фотографування рослин, завантаження зображень у систему для аналізу, отримання детальної інформації про види. Збереження історії ідентифікованих рослин та використання інтерактивних інструментів, таких як можливість додавати коментарі або нотатки до результатів ідентифікації.

**Наукові співробітники/фахівці** – це користувачі з розширеними правами доступу, які можуть вносити свої корективи до результатів ідентифікації. Вони допомагають у вдосконаленні системи, надаючи свої експертні знання для точнішої класифікації рослин або уточнення даних у базі.

*Функції:* перегляд і корекція результатів ідентифікації рослин, додавання нових рослин або характеристик до бази даних і надання зворотного зв'язку для вдосконалення алгоритмів нейронної мережі.

Описано можливі сценарії роботи:

– *редагування облікових записів і призначення ролей.* Адміністратор редагує/видаляє/блокує облікові записи для нових користувачів, визначає їхні ролі та надає відповідні права доступу. Наприклад, науковий співробітник може мати додаткові можливості для редагування інформації про рослини, а звичайний користувач — лише можливість ідентифікації;

– *ідентифікація рослин.* Користувач робить фото рослини через мобільний пристрій та завантажує його до системи. Нейронна мережа аналізує

зображення і надає результати щодо виду рослини. Система надає користувачу всю доступну інформацію про рослину, включаючи її характеристики, регіон поширення та інші деталі;

– *оновлення бази даних.* Адміністратор або науковий співробітник оновлює базу даних, додаючи нові види рослин або виправляючи інформацію про існуючі. У разі появи нових видів рослин або зміни їхніх характеристик, база даних оновлюється для актуалізації інформації;

– *аналітичні дані та звіти.* Система може генерувати звіти щодо активності користувачів, ідентифікованих видів та точності роботи нейронної мережі. Дані можуть використовуватися для вдосконалення системи або проведення наукових досліджень.

Для забезпечення інтуїтивності та зручності використання системи було розроблено макети основних сторінок інтерфейсу, які демонструють ключові функціональні частини застосунку (рис. 1.4–1.5). Макети були створені з урахуванням потреб різних категорій користувачів, включаючи звичайних користувачів, адміністрації та науковців, що дозволяє забезпечити зручну навігацію та ефективну взаємодію з системою. У макетах чітко виокремлено основні елементи інтерфейсу, такі як поля для введення даних, кнопки для взаємодії, а також графічні елементи, що підвищують зручність сприйняття та навігації.

Особливу увагу було приділено адаптивності інтерфейсу. Враховуючи різноманіття пристроїв, на яких користувачі будуть взаємодіяти з застосунком (мобільні телефони та планшети), макети забезпечують коректне відображення елементів інтерфейсу на екранах різних розмірів. Це включає зміну розміщення елементів, зменшення або збільшення їхнього розміру залежно від ширини екрану, а також оптимізацію простору для введення тексту та інших дій.

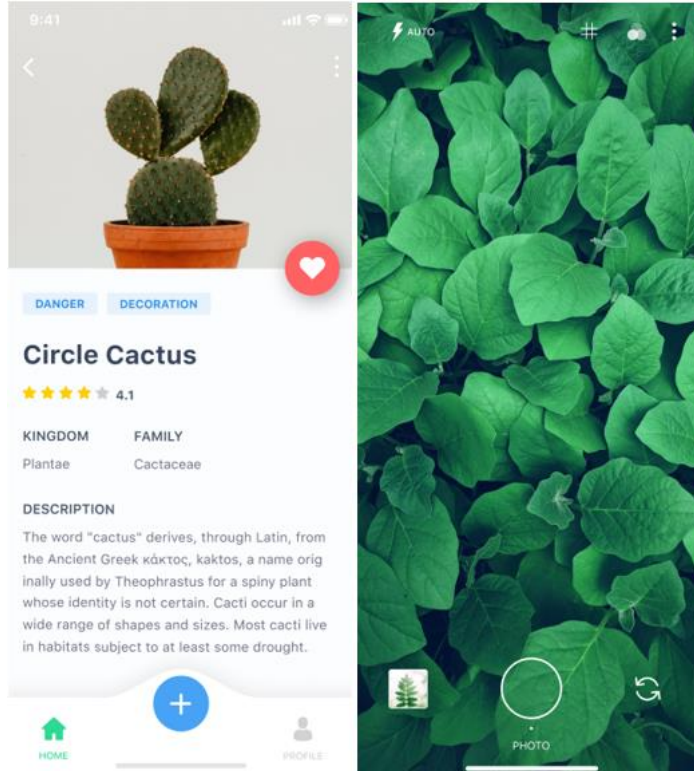


Рисунок 1.4 – Макет сторінок інформації про рослину та завантаження фото рослин

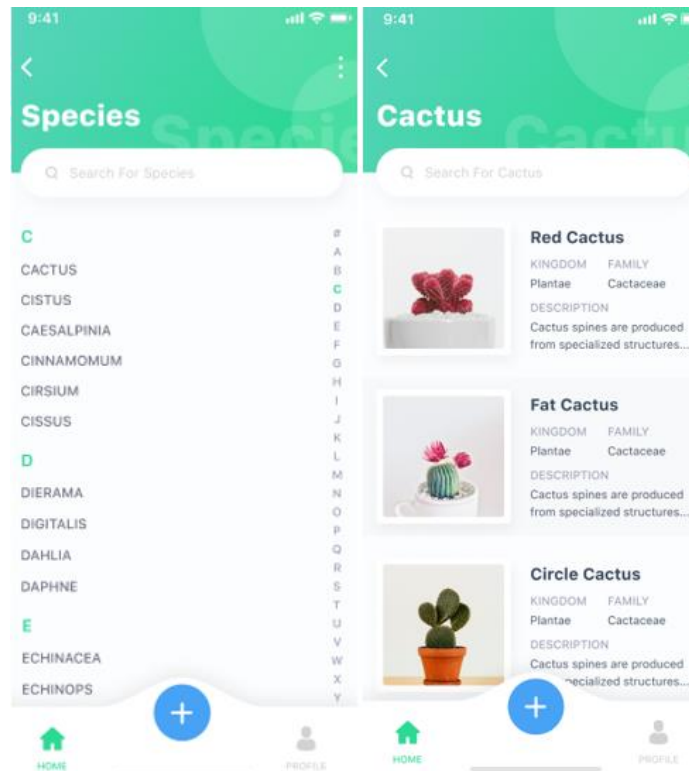


Рисунок 1.5 – Макет пошуку та інформації про рослини

### 1.3 Специфікація вимог застосунка “GreenEye”

**Опис створюваного продукту:** створення застосунку для ідентифікації рослин на основі методів штучного інтелекту є актуальною задачею в умовах зростаючої уваги до екологічних проблем та необхідності популяризації знань про біорізноманіття. Проєкт спрямований на розробку інструменту, який дозволить користувачам ідентифікувати рослини за допомогою штучного інтелекту шляхом аналізу фотографій рослин. Застосунок використовуватиме алгоритми для розпізнавання видів і надання користувачам відповідної інформації про кожен із них. Основним завданням є надання простого та ефективного способу отримання ботанічних знань, що буде корисним як для звичайних користувачів, так і для науковців. Система також дозволить зберігати історію результатів ідентифікації та надавати можливість для подальшого аналізу отриманих даних.

**Призначення продукту:** застосунок “GreenEye” призначений для широкого кола користувачів, включаючи науковців, студентів, екологів, садівників, туристів, ентузіастів природи та звичайних користувачів, які цікавляться рослинним світом. Застосунок дозволяє визначати рослинні види шляхом завантаження зображень і автоматично надає інформацію про ботанічні характеристики, середовище існування та особливості рослин.

#### **Погодження, що ухвалені в програмній документації**

Погоджено, що для розробки застосунку “GreenEye” та його правильного функціонування буде використано технології апаратної та програмної реалізації React Native, бази даних PostgreSQL та відповідних API для інтеграції з екологічними сервісами.

#### **Межі проєкту ПЗ**

Завершення роботи над розробкою ПЗ “GreenEye” – 25.11.2024 р.

#### **Спеціальні вимоги**

1) Застосунок повинен бути доступним для використання на телефонах і планшетах.



- 2) Дані про ідентифіковані рослини повинні бути доступні для перегляду в будь-який час.
- 3) Застосунок повинен мати можливість відслідковувати і зберігати історію результатів ідентифікацій.
- 4) Інтерфейс застосунку повинен бути зручним та інтуїтивно зрозумілим для користувачів з різним рівнем технічної підготовки.

**Опис призначеного для користувача інтерфейсу** має включати такі елементи:

- 1) *головний екран*: простий і зручний інтерфейс із можливістю швидкого доступу до основних функцій (завантаження зображень для ідентифікації, пошуку рослин та налаштувань). Інтуїтивно зрозуміла навігація для легкого переміщення між різними розділами застосунка;
- 2) *екран ідентифікації рослини*: поле для завантаження або фотографування рослини, кнопка для початку процесу ідентифікації, виведення результату з докладною інформацією про рослину (назва, характеристики, середовище існування, а також рекомендації, якщо застосовується);
- 3) *екран результатів ідентифікації*: показ ідентифікованого виду рослини разом із зображенням, опис видів, ботанічні характеристики та додаткова інформація;
- 4) *історія ідентифікацій*: список всіх раніше ідентифікованих рослин із можливістю перегляду кожного результату та додаткової інформації, функції пошуку та фільтрації для зручного доступу до потрібних записів;
- 5) *налаштування користувача*: опції для персоналізації інтерфейсу, можливість налаштувати сповіщення;
- 6) *допомога та підтримка*: розділ із відповідями на часто поставлені запитання (FAQ) та контактами служби підтримки, інтерактивні підказки щодо використання застосунку для нових користувачів.

**План розробки системи ідентифікації рослин:**

- 1) аналіз вимог користувачів та створення технічного завдання;

- 2) розробка архітектури системи та вибір технологій для реалізації;
- 3) розробка бази даних;
- 4) розробка інтерфейсу для користувачів;
- 5) розробка ролі адміністратора, яка дозволяє редагувати профілі користувачів, включаючи фахівців та звичайних користувачів. Адміністратор може керувати доступом до різних функцій застосунку, переглядати активність користувачів, моніторити прогрес у використанні застосунку для ідентифікації рослин, а також отримувати статистику та звіти про активність користувачів;
- б) розробка ролі користувачів, яка дозволяє ідентифікувати рослини за допомогою ШІ, зберігати результати ідентифікації у персональному профілі та мати доступ до бази даних рослин з детальними описами;
- 7) розробка ролі фахівців, яка дозволяє переглядати і змінювати результати ідентифікації рослин, додавати нові рослини або характеристики до бази даних і надавати зворотній зв'язок для вдосконалення алгоритмів нейронної мережі;
- 8) тестування та відлагодження системи;
- 9) оформлення документації.

### **Сфера застосування**

Застосунок для ідентифікації рослин “GreenEye” може бути використаний у різних сферах, де потрібне точне та швидке визначення рослин. Основними напрямками є:

1) *освіта*: застосунок може бути корисним для шкіл, університетів, ботанічних садів та природничих музеїв. Він дозволяє учням і студентам легко ідентифікувати рослини під час практичних занять або польових досліджень, сприяючи покращенню процесу навчання біології та ботаніки;

2) *екологія та природоохоронні дослідження*: застосунок “GreenEye” може допомогти екологам, біологам та ботанікам швидко ідентифікувати рослинні види у польових умовах, що особливо важливо при дослідженні біорізноманіття,

моніторингу змін в екосистемах або під час роботи над проектами з охорони природи;

3) *сільське господарство*: фермерам і агрономам застосунок допомагає визначати рослини на полях, розпізнавати бур'яни та шкідливі види рослин. Це сприяє більш ефективному управлінню посівами та покращенню врожайності, дозволяючи швидше реагувати на зміни в агроландшафті;

4) *ландшафтний дизайн та садівництво*: для ландшафтних дизайнерів і садівників “GreenEye” стане корисним інструментом для ідентифікації рослин під час роботи на об'єктах. Він дозволяє точно визначати види рослин, що сприяє кращому плануванню озеленення територій;

5) *туризм та активний відпочинок*: любителі природи, туристи та мандрівники можуть використовувати застосунок для ідентифікації рослин під час прогулянок у лісах, горах та парках. Це допомагає краще розуміти природу навколо та підвищує інтерес до дослідження флори.

Таким чином, “GreenEye” має широкий спектр застосування у науковій, освітній, аграрній та розважальній сферах. Він робить процес ідентифікації рослин доступним для всіх, від професіоналів до любителів природи, та сприяє покращенню екологічної обізнаності й охороні навколишнього середовища.

### **Характеристики користувачів**

Основними характеристиками користувачів застосунку “GreenEye” є наявність смартфона або планшета з доступом до Інтернету та вбудованою камерою для фотографування рослин.

### **Структура і склад системи**

Для створення застосунку “GreenEye” необхідно мати такі компоненти: сервер для обробки запитів, база даних для збереження інформації про рослини, модулі штучного інтелекту для аналізу зображень та мобільний інтерфейс, через який користувачі взаємодіють із системою.

### **Загальні обмеження**

Для коректної роботи застосунку “GreenEye” необхідно постійне підключення до Інтернету для отримання доступу до бази даних рослин та відправки зображень для обробки штучним інтелектом.

## **ФУНКЦІЇ СИСТЕМИ ІДЕНТИФІКАЦІЇ РОСЛИН НА ОСНОВІ МЕТОДІВ ШТУЧНОГО ІНТЕЛЕКТУ**

### *Функція ідентифікації рослин*

#### **Опис функції**

Функція ідентифікації рослин дозволяє користувачам зробити фотографію рослини, після чого застосунок аналізує зображення за допомогою нейронних мереж і надає інформацію про її вид.

#### **Вхідна і вихідна інформація**

Вхідна інформація – зображення рослини, зроблене користувачем.

Вихідна інформація – результат ідентифікації, включаючи назву рослини, її наукову класифікацію та додаткову інформацію про вид.

#### **Функціональні вимоги**

- 1) Застосунок повинен дозволяти користувачам робити фотографії рослин безпосередньо через інтерфейс програми.
- 2) Застосунок повинен забезпечити швидку обробку зображення та визначення виду рослини за допомогою алгоритмів штучного інтелекту.
- 3) Застосунок має зберігати результати ідентифікації у профілі користувача для перегляду в майбутньому.
- 4) Користувач повинен мати можливість переглянути детальну інформацію про ідентифіковану рослину, включаючи її наукові та популярні назви, екологічні особливості та використання.

### *Функція історії ідентифікацій*

#### **Опис функції**

Функція дозволяє користувачам переглядати історію своїх ідентифікацій, зберігаючи фотографії рослин та результати розпізнавання.

**Вхідна і вихідна інформація**

Вхідна інформація – список ідентифікованих рослин та зображення, які були зроблені користувачем.

Вихідна інформація – історія ідентифікацій із деталями про кожну рослину.

**Функціональні вимоги**

- 1) Система повинна зберігати історію всіх ідентифікованих рослин у профілі користувача.
- 2) Користувач повинен мати можливість переглядати збережені результати ідентифікацій разом із фотографіями.
- 3) Система повинна надавати можливість сортувати та фільтрувати історію за датою або назвою рослини.

*Функція додавання рослини до списку обраних*

**Опис функції**

Функція дозволяє користувачам зберігати ідентифіковані рослини у «список обраних» для швидкого доступу до інформації про них.

**Вхідна і вихідна інформація**

Вхідна інформація – ідентифікована рослина, яку користувач хоче додати до списку обраних.

Вихідна інформація – перелік обраних рослин з можливістю перегляду інформації про них.

**Функціональні вимоги**

- 1) Користувач повинен мати можливість додавати ідентифіковані рослини до списку обраних.
- 2) Система повинна зберігати перелік обраних рослин та забезпечувати доступ до нього через меню програми.
- 3) Користувач повинен мати можливість видаляти рослини зі списку обраних у будь-який момент.

### *Функція редагування інформації про рослину*

#### **Опис функції**

Функція редагування інформації про рослину в застосунку “GreenEye” дозволяє адміністраторам або експертам оновлювати існуючі дані про рослини. Це необхідно для забезпечення актуальності інформації, додавання нових характеристик або коригування неточних відомостей.

#### **Вхідна і вихідна інформація**

Вхідна інформація – назва рослини, наукова класифікація, додаткові характеристики (опис, екологічні властивості), зображення.

Вихідна інформація – оновлені дані про рослину, підтвердження успішного збереження змін.

#### **Функціональні вимоги**

- 1) Застосунок повинен забезпечувати доступ до інтерфейсу редагування даних тільки для користувачів з відповідними правами (адміністратори або експерти).
- 2) Система повинна дозволяти змінювати текстові та графічні дані про рослини, такі як опис, зображення, характеристики або наукова класифікація.
- 3) Система повинна зберігати історію змін для кожної рослини, щоб у разі необхідності можна було відновити попередні версії даних.
- 4) Застосунок повинен надавати можливість додавання нових зображень та зміни наявних у базі даних.
- 5) Після редагування інформація повинна бути автоматично оновлена і відобразитися користувачам у застосунку.
- 6) Можливість збереження змін та публікація оновлених даних для доступу всім користувачам застосунку.

### **ВИМОГИ ДО ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ**

#### **Джерела і зміст вхідної інформації (даних)**

Основними джерелами вхідної інформації для застосунку “GreenEye” є користувачі, які взаємодіють із системою через смартфони або планшети.

Користувачі можуть робити знімки рослин та завантажувати їх до застосунку для ідентифікації за допомогою штучного інтелекту. Вхідна інформація включає зображення рослин, які обробляються нейронними мережами для визначення їх виду.

Важливо забезпечити захист персональних даних користувачів та конфіденційність оброблюваної інформації, а також стабільність і швидкість роботи системи.

### **Нормативно-довідкова інформація (класифікатори, довідники тощо)**

Для коректної роботи застосунку “GreenEye” необхідні такі довідники та класифікатори:

1) *класифікатори рослин*. Вони допомагають системі правильно класифікувати рослини на основі оброблених зображень. Ці класифікатори містять наукові назви, таксономічні дані та загальні характеристики видів;

2) *довідники з ботанічних термінів*. Ці довідники допомагають користувачам зрозуміти спеціалізовані терміни, що використовуються при описі рослин;

3) *екологічні та географічні довідники*. Використовуються для більш точної ідентифікації рослин, враховуючи регіональні особливості флори;

4) *довідники з охорони природи*. Включають інформацію про охоронювані види рослин, що дозволяє користувачам дізнатися про стан збереження і можливі загрози для виду.

### **Вимоги до способів організації, збереження та ведення інформації**

1) *База даних*. Для збереження інформації про рослини та користувачів застосовується реляційна база даних **PostgreSQL**, яка забезпечує структурування і надійне збереження даних.

2) *RESTful API*. Для обробки запитів між застосунком і серверною частиною використовується RESTful API, що дозволяє забезпечити швидкий обмін даними та обробку зображень.

3) *React Native*. Інтерфейс застосунку розроблений з використанням React Native, що дозволяє створювати кросплатформний мобільний застосунок для Android та iOS. Це забезпечує зручну та ефективну взаємодію користувачів із системою, а також покращує продуктивність застосунка на різних платформах.

### **ВИМОГИ ДО ТЕХНІЧНОГО ЗАБЕЗПЕЧЕННЯ**

Для коректної роботи застосунку “GreenEye” необхідно забезпечити такі технічні вимоги:

1) *мобільні пристрої*. Застосунок повинен працювати на смартфонах та планшетах з операційними системами Android (версія 7.0 і вище) та iOS (версія 12.0 і вище);

2) *процесор*. Пристрої повинні мати достатню обчислювальну потужність для обробки зображень і роботи з алгоритмами штучного інтелекту. Рекомендований мінімум – чотириядерні процесори;

3) *оперативна пам'ять*. Для стабільної роботи застосунка необхідно щонайменше 2 ГБ оперативної пам'яті;

4) *інтернет-з'єднання*. Стабільне підключення до Інтернету потрібно для завантаження зображень на сервер та обробки даних;

5) *камера*. Пристрої повинні мати камеру з мінімальною роздільною здатністю 8 Мп для якісного фотографування рослин.

### **ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

#### **Архітектура програмної системи**

Система складається з трьох основних компонентів: програмного забезпечення на стороні користувача, програмного забезпечення на серверній стороні та бази даних.

#### **Системне програмне забезпечення**

1) *Клієнтська частина*. Мобільний застосунок розроблений на основі React Native, що дозволяє реалізувати кросплатформну розробку для Android і iOS.

2) *Серверна частина*. Для back-end частини використовується Nest.js, що працює разом із PostgreSQL для обробки та збереження даних.



## **Мережне програмне забезпечення**

Для розробки застосунку використовувалася операційна система Windows 11, а також інструменти:

Редактор коду — **Visual Studio Code**.

Тестування застосунку — Google Chrome для браузерної частини та емулятори для мобільних платформ.

## **Програмне забезпечення ведення інформаційної бази**

Для управління даними застосунку використовується база даних PostgreSQL. Взаємодія з базою здійснюється через операції створення, читання, оновлення та видалення даних (CRUD).

## **Мова і технологія розробки ПЗ**

Основною мовою розробки є JavaScript. Для клієнтської частини застосовується React Native, а для серверної частини — Nest.js.

## **ВИМОГИ ДО ЗОВНІШНІХ ІНТЕРФЕЙСІВ**

### **Інтерфейс користувача**

Нижче подано вимоги до зовнішніх інтерфейсів користувача застосунку “GreenEye”.

- 1) *Простий і зрозумілий інтерфейс* для користувачів з різними рівнями технічної грамотності, що дозволяє легко ідентифікувати рослини.
- 2) *Інтуїтивно зрозуміле розташування функцій* для швидкого доступу до основних інструментів, таких як завантаження фото та перегляд результатів.
- 3) *Уніфікований стиль оформлення* для всіх екранів застосунку, що створює узгоджений досвід користувача.
- 4) *Адаптивний дизайн*, що забезпечує комфортне користування на смартфонах і планшетах з різними розмірами екранів.
- 5) *Персоналізація інтерфейсу*, що дозволяє користувачам налаштовувати відображення інформації (наприклад, обране або історія ідентифікацій).
- 6) *Зручна навігація*, що дозволяє швидко переходити між основними розділами: фото, результати, історія ідентифікацій.

- 7) *Функція пошуку* та можливість фільтрувати результати за категоріями (наприклад, рослини, які ідентифіковано).
- 8) *Мінімальна кількість кроків* для виконання основних дій, таких як завантаження фото та отримання результатів.
- 9) *Захист персональних даних*, що забезпечує конфіденційність інформації користувачів і фотографій рослин.

### **Апаратний інтерфейс**

Апаратний інтерфейс включає мобільні пристрої користувачів — смартфони або планшети, оснащені камерами, що дозволяють робити знімки рослин і завантажувати їх для аналізу.

### **Програмний інтерфейс**

**React Native** — це популярний фреймворк для розробки мобільних застосунків, який дозволяє створювати кросплатформні рішення для Android та iOS.

Компоненти React Native забезпечують динамічний інтерфейс, який реагує на взаємодії користувачів і дозволяє ефективно обробляти зображення рослин у реальному часі. Завдяки використанню JavaScript та React, розробники можуть швидко створювати інтуїтивно зрозумілі інтерфейси, які легко налаштовуються під потреби користувачів.

Крім того, React Native підтримує гарячу перезагрузку, що значно прискорює процес розробки, дозволяючи розробникам миттєво бачити зміни без втрати стану застосунку. Це робить його ідеальним вибором для створення мобільних застосунків, які вимагають високої продуктивності та зручності в користуванні.

### **ВЛАСТИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

1) *Модульність*. Архітектура застосунку “GreenEye” побудована за модульним принципом, що дозволяє легко розширювати його функціональність (наприклад, додавання нових функцій або алгоритмів штучного інтелекту).

- 2) *Множинна платформа.* Застосунок розроблений для роботи на мобільних пристроях із операційними системами Android та iOS завдяки React Native.
- 3) *Безпека.* Забезпечується механізми автентифікації та шифрування для захисту даних користувачів і фотографій рослин.
- 4) *Можливості звітності.* Застосунок веде історію ідентифікацій, надає можливість переглядати статистику і зберігати результати.
- 5) *Керування користувачами.* Застосунок дозволяє користувачам створювати власні профілі, переглядати історію ідентифікацій та зберігати обрані рослини.
- 6) *Функціональність.* Основні функції включають завантаження фото, ідентифікацію рослин, перегляд інформації про рослини та ведення історії ідентифікацій.
- 7) *Легкість використання.* Інтерфейс інтуїтивно зрозумілий і не потребує складної підготовки для роботи з застосунком, забезпечуючи швидке і зручне виконання основних завдань.

Всі ці властивості є важливими для забезпечення ефективної роботи застосунку “GreenEye” та його популярності серед широкого кола користувачів.

#### **1.4 Аналіз методів штучного інтелекту для системи**

У сучасних розробках мобільних застосунків, які використовують штучний інтелект (ШІ), існує кілька методів і алгоритмів, які можуть бути ефективно застосовані для розв’язання завдань ідентифікації рослин. У даному контексті розглянемо кілька ключових підходів, які можуть бути реалізовані у застосунку ідентифікації рослин [5].

*Класифікація* – алгоритми машинного навчання, такі як метод опорних векторів (SVM) та дерева рішень, можуть бути використані для класифікації рослин на основі їх характеристик (наприклад, форми листя, кольору квітів). Ці алгоритми

навчаються на еталонних даних, що дозволяє їм ідентифікувати рослини в нових зображеннях з високою точністю.

*Регресійні моделі* можуть бути застосовані для прогнозування певних параметрів рослин, таких як їх вік або потреби у воді, на основі наявних даних. Це дозволяє отримати додаткову інформацію, яка може бути корисною для користувачів застосунку.

*Глибоке навчання*, зокрема конволюційні нейронні мережі (CNN), є основним методом, використовуваним у застосунку. CNN продемонстрували високу ефективність у задачах обробки зображень, завдяки їх здатності автоматично виділяти важливі ознаки рослин із фотографій. Ці мережі складаються з кількох шарів, які дозволяють навчитися розпізнавати складні патерни в зображеннях, що критично важливо для точного ідентифікаційного процесу.

*Методи комп'ютерного бачення* дозволяють автоматично виявляти рослини на основі їхніх зображень. Цей процес включає в себе обробку та аналіз зображень, що дає змогу виявити та класифікувати рослини в різних умовах.

Додатково, *техніки відстеження об'єктів* можуть бути використані для моніторингу рослин у природному середовищі, що дозволяє відстежувати зміни їх стану та розвитку.

*Процес виділення ключових точок* дозволяє виявляти та класифікувати важливі характеристики рослин, такі як форма, текстура та колір. Використання алгоритмів для екстракції ознак допомагає підвищити точність класифікації в рамках застосунку.

Вибір методів для реалізації системи має критичне значення для його успішної роботи. Зосередження на конволюційних нейронних мережах (CNN) забезпечить високий рівень точності в ідентифікації рослин за їх зображеннями, що, в свою чергу, сприятиме розвитку користувацького досвіду та наданню цінної інформації користувачам.

## **Висновки до розділу 1**

В першому розділі проведено аналіз предметної області, зокрема вивчення існуючих застосунків для ідентифікації рослин на основі штучного інтелекту. Розглянуто й проаналізовано кілька аналогічних застосунків, що функціонують у цій сфері. Визначено специфікацію вимог до програмного забезпечення, що розробляється.

Описано створюваний продукт та його призначення. Зазначено погодження, що були ухвалені в програмній документації та межі проєкту ПЗ. Встановлено спеціальні вимоги до системи і детально описано інтерфейс, призначений для користувача.

Сформовано план розробки застосунку та окреслено сферу його застосування. Визначено характеристики користувачів, структуру і склад системи, а також описано загальні обмеження та функціонал застосунку.

Встановлено вимоги до інформаційного забезпечення, джерела та зміст вхідної інформації, нормативно-довідкову інформацію, а також вимоги до способів організації, збереження та ведення даних. Зазначено вимоги до технічного забезпечення, програмного забезпечення, описано архітектуру програмної системи, системне програмне забезпечення, мережне програмне забезпечення та програмне забезпечення для ведення інформаційної бази.

Визначено мову та технологію розробки ПЗ, вимоги до зовнішніх інтерфейсів, зокрема інтерфейс користувача, апаратний інтерфейс, програмний інтерфейс, а також описано комунікаційний протокол та властивості програмного забезпечення.

Описано різні методи штучного інтелекту, які можуть бути використані для ідентифікації рослин, включаючи методи машинного навчання, глибокого навчання та комп'ютерного зору. Зокрема, акцентовано увагу на методах глибокого навчання, оскільки вони демонструють найкращі результати у розпізнаванні образів та можуть бути ефективно застосовані в розробці мобільного застосунку.

## 2 МОДЕЛЮВАННЯ СИСТЕМИ ІДЕНТИФІКАЦІЇ РОСЛИН

### 2.1 Етапи реалізації проєкту

Реалізація проєкту моделювання системи ідентифікації рослин на основі штучного інтелекту може бути поділена на кілька етапів.

1) *Аналіз вимог та потреб.* На цьому етапі проводиться дослідження ринку, щоб визначити потреби користувачів, включаючи ботаніків, екологів, туристів, природоохоронців та інших ентузіастів природи. Важливо з'ясувати, які функції має виконувати система, які види рослин вона повинна ідентифікувати, а також які додаткові можливості можуть бути корисними.

2) *Проектування архітектури системи.* На цьому етапі створюються детальні плани структури застосунку, обираються необхідні технології для обробки зображень та машинного навчання, а також розробляються сценарії роботи системи.

3) *Розробка та тестування.* Після проектування відбувається розробка програмного коду, зокрема алгоритмів штучного інтелекту, а також інтерфейсу користувача. Після завершення розробки проводиться тестування системи, щоб перевірити її точність та відповідність вимогам.

4) *Впровадження та налаштування.* На цьому етапі застосунок встановлюється на сервері (якщо потрібно) та налаштовується для роботи з користувачами. Важливо впевнитися, що система функціонує стабільно та забезпечує коректну ідентифікацію рослин.

5) *Підтримка та розвиток.* Після впровадження застосунку необхідно забезпечити його підтримку, включаючи вирішення технічних проблем, оновлення бази даних рослин та вдосконалення алгоритмів. Також важливо враховувати зворотний зв'язок від користувачів для покращення функціоналу.

Крім того, реалізація проєкту може зайняти значний час та вимагати значних фінансових і людських ресурсів. Тому на кожному етапі важливо забезпечувати

ефективне управління проектом, встановлювати реалістичні терміни та бюджет, а також використовувати методики управління проектами.

Усі ці етапи та вимоги мають бути враховані при реалізації проекту моделювання системи ідентифікації рослин, щоб забезпечити успішну та ефективну реалізацію проекту та задоволення потреб користувачів.

Під час проектування архітектури застосунку програмне забезпечення розробляється шляхом створення різних типів UML-діаграм, таких як діаграми варіантів використання, діаграми послідовності, діаграми класів, станів та діяльності, використовуючи спеціалізовані продукти, такі як draw.io.

Створення UML-діаграм допомагає розробникам краще зрозуміти структуру проекту. Після цього наступним етапом є розробка мокапів застосунку відповідно до вимог, включаючи логотип і вибір кольорової гами.

Після цього відбувається розробка front-end та back-end частин застосунку, а також тестування на помилки для запобігання проблем при роботі програмного забезпечення.

## **2.2 Створення USE CASE**

*Сценарії використання (use cases)* є важливим інструментом для визначення вимог до програмного забезпечення, оскільки вони описують функціональність системи з перспективи користувача. Вони ілюструють, яким чином користувачі взаємодіють із застосунком і як система повинна реагувати на їх дії [6].

Кожен сценарій використання містить чіткий перелік дій, які користувач повинен виконати для досягнення конкретної мети, а також результати, які система має надати у відповідь. Ці сценарії не лише допомагають у розробці та тестуванні програмного забезпечення, але й сприяють кращому розумінню вимог серед усіх учасників процесу — розробників, тестувальників та замовників [7].

Сценарії використання можуть включати різні ролі користувачів, такі як звичайні користувачі, фахівці та адміністратори, кожен із яких має свої специфічні

потреби та очікування. Це дозволяє створити більш точний і функціональний продукт, що задовольняє вимоги різних груп користувачів.

### **Короткий сценарій використання**

Користувач відкриває застосунок для ідентифікації рослин і робить фотографію рослини. Після аналізу зображення система виводить результати ідентифікації, включаючи назву рослини, її опис та інформацію про догляд. Коли користувач отримує необхідну інформацію, він може зберегти результати або поділитися ними через соціальні мережі. Система працює коректно, запити виконуються задовільно. Коли мета користувача була досягнута, він виходить із системи.

### **Поверхневий сценарій використання «Ідентифікація рослини»**

Головний сценарій (успішний):

- 1) користувач відкриває застосунок і надає доступ до камери;
- 2) користувач робить фотографію рослини;
- 3) система обробляє зображення та аналізує його за допомогою алгоритмів штучного інтелекту;
- 4) система відображає інформацію про ідентифіковану рослину, включаючи її назву, опис та рекомендації щодо догляду;
- 5) користувач може зберегти інформацію в особистий профіль або поділитися нею в соціальних мережах;
- 6) користувач може повторити процес для іншої рослини або закрити застосунок.

### **Альтернативні сценарії**

- 1) Система не може ідентифікувати рослину через низьку якість зображення.
- 2) Користувач надає неправильні дані (наприклад, намагається завантажити зображення не пов'язане з рослинами), і система видає повідомлення про помилку.



3) У випадку технічного збою (наприклад, немає з'єднання з інтернетом) система видає повідомлення «Немає зв'язку з сервером», і користувач може написати у службу підтримки.

4) Користувач намагається ідентифікувати рідкісну рослину, і система не має достатньої інформації для її ідентифікації, на що користувач отримує відповідне повідомлення.

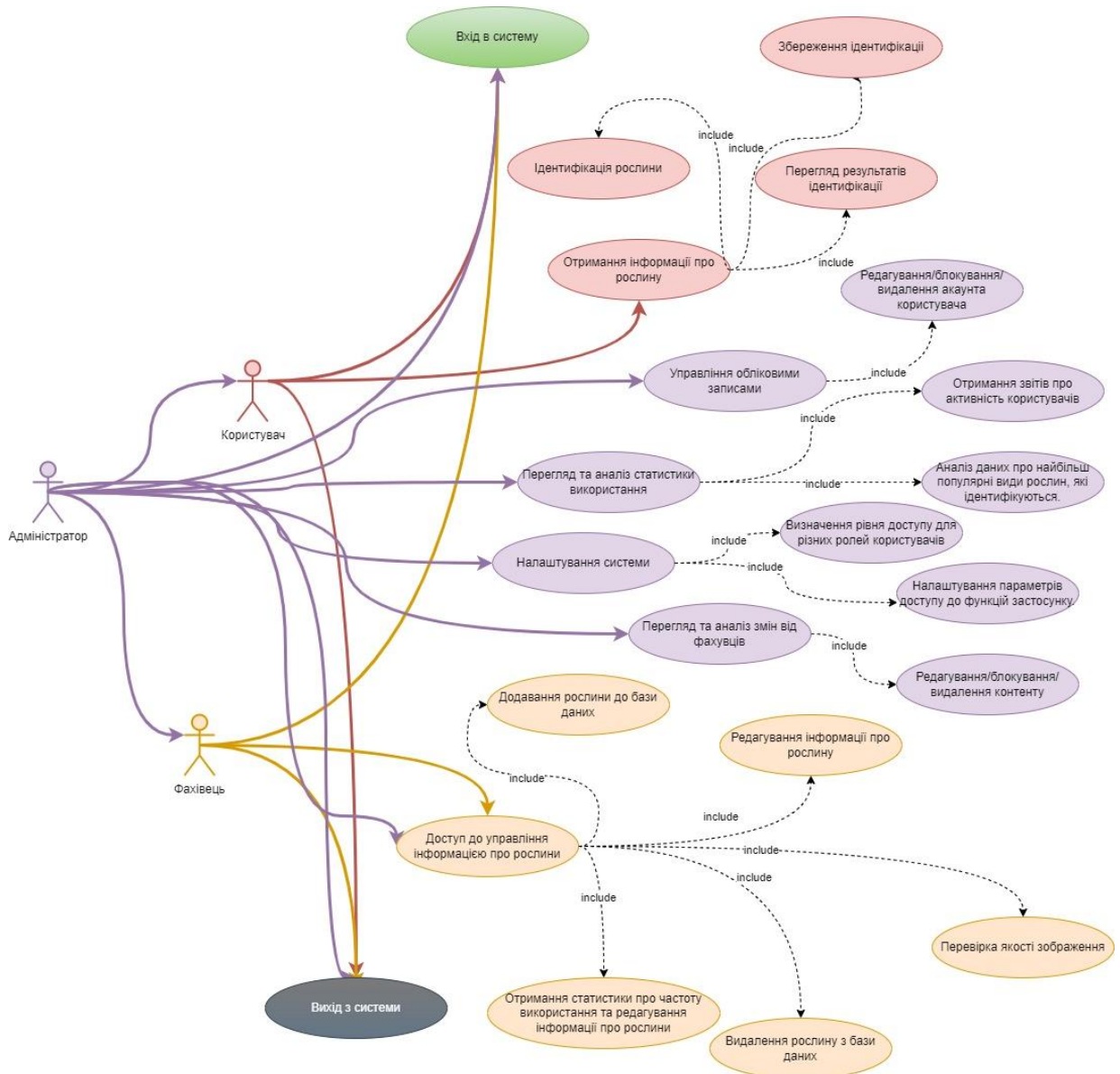


Рисунок 2.1 – Діаграма використання системи ідентифікації рослин

Діаграма використання системи ідентифікації рослин ілюструє взаємодію користувачів із ключовими функціями застосунку. Представлений сценарій є основою для подальшого моделювання та реалізації системи.

### Опис дійових осіб

*Адміністратор* – особа, відповідальна за управління застосунком і забезпечення його належної роботи. Адміністратор контролює доступ до системи, управляє користувачами, а також відповідає за безпеку даних.

*Користувач* – особа, яка використовує застосунок для ідентифікації рослин. Користувачі можуть бути любителями рослин, студентами, науковцями або будь-якими іншими особами, які цікавляться ботанікою.

*Науковий співробітник/фахівець* – професіонал з експертними знаннями в галузі ботаніки, відповідальний за підтримку та розвиток бази даних рослин.

### Опис варіантів використання. Варіант використання «Вхід в систему»

**Передумови:** користувач має активний обліковий запис у системі.

**Постумови:** якщо варіант використання виконаний успішно, користувач входить в систему. В іншому випадку стан системи не змінюється.

Таблиця 2.1 – Головний розділ сценарію виконання варіанта використання «Вхід в систему»

Use case section	Comment
Use Case Name	Увійти до системи
Scope	Мобільний застосунок для ідентифікації рослин
Level	Успішно пройти процес авторизації в системі, отримати доступ до власного профілю.
Primary Actor	Користувач, науковий співробітник/фахівець, адміністратор.
Stakeholders and interests	1) Користувач: зацікавлений у швидкому доступі до інформації про рослини. 2) Науковий співробітник/фахівець: зацікавлений у використанні застосунку для досліджень. 3) Адміністратор: зацікавлений у забезпеченні належного функціонування системи та її підтримці.

## Кінець таблиці 2.1

<b>Preconditions</b>	<p>1) Користувач, науковий співробітник/фахівець, адміністратор мають бути зареєстровані, їх профіль має бути активований.</p> <p>2) Застосунок встановлений на мобільному пристрої користувача, наукового співробітника/фахівця, адміністратора.</p> <p>3) Користувач, науковий співробітник/фахівець, адміністратор знають свої облікові дані (електронна пошта та пароль).</p>
<b>Success guarantee</b>	<p>1) Користувач, науковий співробітник/фахівець, адміністратор використовують мобільний пристрій, що підтримує застосунок.</p> <p>2) Користувач, науковий співробітник/фахівець, адміністратор мають стабільне підключення до Інтернету.</p> <p>3) Користувач, науковий співробітник/фахівець, адміністратор вводять коректні дані профілю (електронна пошта та пароль).</p>
<b>Main Success Scenario</b>	<p>1) Користувач, науковий співробітник/фахівець, адміністратор відкривають застосунок.</p> <p>2) Користувач, науковий співробітник/фахівець, адміністратор вводять свою електронну пошту та пароль у спеціалізовані поля.</p> <p>3) Користувач, науковий співробітник/фахівець, адміністратор натискають кнопку «Увійти».</p> <p>4) Система перевіряє облікові дані</p> <p>5) У разі успішної перевірки, система перенаправляє користувача, наукового співробітника/фахівця або адміністратора на головний екран застосунку.</p>

**Альтернативні сценарії**

1) Користувач намагається увійти в систему, але не має зареєстрованого облікового запису. Система відображає повідомлення про помилку та пропонує зареєструватися.

2) Користувач намагається увійти в систему, але не має підключення до Інтернету. Система відображає повідомлення про відсутність з'єднання та пропонує перевірити налаштування мережі.

3) Користувач намагається увійти в систему, використовуючи застарілу версію операційної системи Android або iOS. Система не підтримує цю версію та

пропонує користувачеві оновити операційну систему до останньої версії для коректної роботи застосунку.

4) Користувач намагається увійти в систему, але швидкість доступу до Інтернету дуже низька. Система може працювати некоректно або повільно реагувати, і відображає повідомлення про проблеми зі з'єднанням.

### Варіант використання «Ідентифікація рослини»

**Передумови:** користувач має активний обліковий запис у системі.

**Постумови:** якщо варіант використання виконано успішно, користувач отримує детальну інформацію про ідентифіковану рослину. В іншому випадку, система надає повідомлення про невдачу.

Таблиця 2.2 – Головний розділ сценарію виконання варіанта використання «Ідентифікація рослини»

Варіант використання	Ідентифікація рослини
<b>Актори</b>	Користувач
<b>Короткий опис</b>	Ідентифікація рослини за допомогою камери мобільного пристрою
<b>Мета</b>	Отримати інформацію про рослину
<b>Тип</b>	Підлеглий
<b>Посилання на інші варіанти використання</b>	Виконується в рамках варіанту використання «Отримання інформації про рослину»

Таблиця 2.3 – Типовий хід подій сценарію виконання варіанта використання «Ідентифікація рослини»

Дії актора	Відгук системи
1) Користувач відкриває застосунок.	2) Система відображає головний екран застосунку.
3) Користувач вибирає опцію «Ідентифікація рослини».	4) Система показує інструкції для фотографування рослини.
5) Користувач фотографує рослину. <b>Виняток №1.</b> Користувач завантажує розміту фотографію.	6) Система обробляє зображення. 7) Система проводить ідентифікацію. 8) Система відображає інформацію про рослину, включаючи її назву, характеристики та рекомендації.

Таблиця 2.4 – Винятки сценарію виконання варіанта використання «Ідентифікація рослини»

Дії актора	Відгук системи
<b>Виняток №1.</b> Користувач завантажує розміту фотографію.	
3) Користувач фотографує рослину.	4) Система не може провести ідентифікацію через низьку якість зображення.
5) Користувач повертається до пункту 4 типового ходу подій.	

### Варіант використання «Додавання рослини до бази даних»

**Передумови:** користувач повинен бути авторизованим у системі.

**Постумови:** якщо варіант використання виконано успішно, інформація про рослину буде додана до бази даних. В іншому випадку, стан системи не змінюється.

Таблиця 2.5 – Головний розділ сценарію виконання варіанта використання «Додавання рослини до бази даних»

<b>Варіант використання</b>	Додавання рослини до бази даних
<b>Актори</b>	Науковий співробітник/фахівець
<b>Короткий опис</b>	Додавання нової рослини до бази даних
<b>Мета</b>	Додати інформацію про рослину
<b>Тип</b>	Підлеглий
<b>Посилання на інші варіанти використання</b>	Виконується в рамках варіанту використання «Доступ до управління інформацією про рослини».

Таблиця 2.6 – Типовий хід подій сценарію виконання варіанта використання «Додавання рослини до бази даних»

Дії актора	Відгук системи
1) Науковий співробітник/фахівець обирає дію «Додати рослину».	2) Система виводить форму для введення даних рослини.
3) Науковий співробітник вводить назву рослини та її характеристики.	4) Система перевіряє коректність введених даних.
5) Науковий співробітник підтверджує додавання. <b>Виняток №1.</b> Введені дані не пройшли валідацію.	6) Система зберігає інформацію про рослину в базі даних.

Таблиця 2.7 – Винятки сценарію виконання варіанта використання «Додавання рослини до бази даних»

Дії актора	Відгук системи
<b>Виняток №1.</b> Введені дані не пройшли валідацію.	
5) Науковий співробітник підтверджує додавання.	б) Система повідомляє про помилку в даних.
7) Науковий співробітник повертається до пункту 3 типового ходу подій.	

### Варіант використання «Редагування інформації про рослину»

**Передумови:** користувач повинен бути зареєстрований і володіти необхідними правами.

**Постумови:** якщо варіант використання виконаний успішно, зміни, внесені спеціалістом, будуть збережені в базі даних. В іншому випадку стан системи не змінюється.

Таблиця 2.8 – Головний розділ сценарію виконання варіанта використання «Редагування інформації про рослину»

<b>Варіант використання</b>	Редагування інформації про рослину
<b>Актори</b>	Науковий співробітник/фахівець
<b>Короткий опис</b>	Зміна інформації про рослину в базі даних.
<b>Мета</b>	Оновити дані про рослину
<b>Тип</b>	Підлеглий
<b>Посилання на інші варіанти використання</b>	Виконується в рамках «Доступ до управління інформацією про рослини».

Таблиця 2.9 – Типовий хід подій сценарію виконання варіанта використання «Редагування інформації про рослину»

Дії актора	Відгук системи
1) Науковий співробітник/фахівець обирає рослину для редагування.	2) Система виводить форму з даними про рослину.
3) Науковий співробітник/фахівець вносить зміни.	5) Система перевіряє коректність введених даних.
4) Науковий співробітник/фахівець підтверджує збереження змін.	б) Система повідомляє про успішне оновлення.
<b>Виняток №1.</b> Користувач вводить некоректні дані.	

Таблиця 2.10 – Винятки сценарію виконання варіанта використання «Редагування інформації про рослину»

Дії актора	Відгук системи
<b>Виняток №1.</b> Науковий співробітник/фахівець вводить некоректні дані.	
3) Науковий співробітник/фахівець підтверджує збереження змін.	4) Система виводить повідомлення про помилку, збереження не відбувається.
5) Науковий співробітник/фахівець повертається до пункту 3 типового ходу подій.	

### Варіант використання «Видалення рослини з бази даних»

**Передумови:** користувач повинен бути зареєстрований і володіти необхідними правами.

**Постумови:** якщо варіант використання виконаний успішно, рослина буде видалена з бази даних. В іншому випадку стан системи не змінюється.

Таблиця 2.11 – Головний розділ сценарію виконання варіанта використання «Видалення рослини з бази даних»

<b>Варіант використання</b>	Видалення рослини з бази даних
<b>Актори</b>	Науковий співробітник/фахівець
<b>Короткий опис</b>	Видалення інформації про рослину з бази даних.
<b>Мета</b>	Видалити рослину з системи.
<b>Тип</b>	Підлеглий
<b>Посилання на інші варіанти використання</b>	Виконується в рамках варіанту використання «Доступ до управління інформацією про рослини».

Таблиця 2.12 – Типовий хід подій сценарію виконання варіанта використання «Видалення рослини з бази даних»

Дії актора	Відгук системи
1) Науковий співробітник/фахівець обирає рослину для видалення.	2) Система запитує підтвердження видалення.
3) Науковий співробітник/фахівець підтверджує видалення.	4) Система видаляє рослину з бази даних.
<b>Виняток №1.</b> Рослина не знайдена в базі даних.	5) Система повідомляє про успішне видалення.

Таблиця 2.13 – Винятки сценарію виконання варіанта використання «Видалення рослини з бази даних»

Дії актора	Відгук системи
<b>Виняток №1.</b> Рослина не знайдена в базі даних.	
1) Науковий співробітник/фахівець підтверджує видалення.	2) Система виводить попередження про те, що рослина не знайдена.
3) Науковий співробітник/фахівець повертається до пункту 1 типового ходу подій.	

### Варіант використання «Збереження ідентифікації рослини»

**Передумови:** користувач успішно ідентифікував рослину.

**Постумови:** якщо варіант використання виконаний успішно, інформація про ідентифікацію буде збережена в профілі користувача. В іншому випадку стан системи не змінюється.

Таблиця 2.14 – Головний розділ сценарію виконання варіанта використання «Збереження ідентифікації рослини»

<b>Варіант використання</b>	Збереження ідентифікації рослини
<b>Актори</b>	Користувач
<b>Короткий опис</b>	Збереження інформації про ідентифіковану рослину в особистому кабінеті користувача.
<b>Мета</b>	Зберегти результати ідентифікації.
<b>Тип</b>	Підлеглий
<b>Посилання на інші варіанти використання</b>	Виконується в рамках варіанту використання «Отримання інформації про рослину».

Таблиця 2.15 – Типовий хід подій сценарію виконання варіанта використання «Збереження ідентифікації рослини»

Дії актора	Відгук системи
1) Користувач обирає опцію збереження ідентифікації.	2) Система запитує підтвердження збереження.
3) Користувач підтверджує збереження. <b>Виняток №1.</b> Користувач вже зберіг цю ідентифікацію.	4) Система зберігає дані в профілі користувача. 5) Система повідомляє про успішне збереження.



Таблиця 2.16 – Винятки сценарію виконання варіанта використання «Збереження ідентифікації рослини»

Дії актора	Відгук системи
<b>Виняток №1.</b> Користувач вже зберіг цю ідентифікацію.	
3) Користувач підтверджує збереження.	4) Система перевіряє наявність збережених даних.
6) Користувач повертається до пункту 1 типового ходу подій.	5) Система виводить попередження про те, що дані вже збережені.

### Варіант використання «Видалення користувача з бази даних»

**Передумови:** адміністратор повинен бути зареєстрований і володіти необхідними правами.

**Постумови:** якщо варіант використання виконаний успішно, користувач буде видалений з бази даних. В іншому випадку стан системи не змінюється.

Таблиця 2.17 – Головний розділ сценарію виконання варіанта використання «Видалення користувача з бази даних»

<b>Варіант використання</b>	Видалення користувача з бази даних
<b>Актори</b>	Адміністратор
<b>Короткий опис</b>	Видалення інформації про користувача з бази даних.
<b>Мета</b>	Видалити користувача з системи.
<b>Тип</b>	Підлеглий
<b>Посилання на інші варіанти використання</b>	Виконується в рамках варіанту використання «Управління обліковими записами».

Таблиця 2.18 – Типовий хід подій сценарію виконання варіанта використання «Видалення користувача з бази даних»

Дії актора	Відгук системи
1) Адміністратор обирає користувача для видалення. <b>Виняток №1.</b> Користувача не знайдено в базі даних.	2) Система запитує підтвердження видалення.
3) Адміністратор підтверджує видалення.	4) Система видаляє користувача з бази даних. 5) Система повідомляє про успішне видалення.

Таблиця 2.19 – Винятки сценарію виконання варіанта використання «Видалення рослини з бази даних»

Дії актора	Відгук системи
<b>Виняток №1.</b> Користувача не знайдено в базі даних.	
1) Адміністратор обирає користувача для видалення.	2) Система виводить попередження про те, що користувача не знайдено.
3) Адміністратор повертається до пункту 1 типового перебігу подій.	

Загалом, варіанти використання описують функціональність застосунку для ідентифікації рослин. Кожен сценарій охоплює дії користувачів, такі як ідентифікація, додавання, редагування, видалення рослин, управління обліковими записами тощо. Це дозволяє чітко зрозуміти взаємодію між акторами і системою, що сприяє поліпшенню користувацького досвіду та ефективності застосунку.

### 2.3 Алгоритм роботи програмного застосунка

Алгоритм роботи ПЗ застосунку для ідентифікації рослин може бути наступним:

- 1) *реєстрація користувачів.* Нові користувачі повинні зареєструватися в системі, заповнивши необхідну інформацію, таку як ім'я, електронна пошта та пароль;
- 2) *авторизація користувачів.* Користувачі повинні авторизуватися в системі, ввівши свій логін та пароль;
- 3) *доступ до бази даних рослин.* Після авторизації користувач має доступ до переліку рослин, які можна ідентифікувати, а також до додаткової інформації про них;
- 4) *ідентифікація рослин.* Користувач може зробити фото рослини або завантажити його з галереї. Система аналізує зображення та надає результати ідентифікації, включаючи назву рослини та опис;

- 5) *додавання рослини до бази даних.* Фахівці можуть додавати нові рослини до бази даних, заповнюючи форму з інформацією про рослину, включаючи фото та опис;
- 6) *моніторинг активності.* Система може відстежувати активність користувачів, надаючи статистику про кількість ідентифікацій, доданих рослин і запитів до фахівців;
- 7) *оцінювання.* Користувачі можуть оцінювати точність ідентифікації та надану інформацію, що допомагає покращити систему;
- 8) *адміністрування.* Адміністратори мають доступ до управління користувачами, рослинами та базою даних, а також до вирішення проблем, які можуть виникнути під час використання застосунку.

Для графічного зображення алгоритмів і процесів у системі ідентифікації рослин часто використовують блок-схеми, які ілюструють кожен етап взаємодії користувача з застосунком, а також різноманітні варіанти розвитку подій. Діаграма діяльності є альтернативою блок-схемі і відображає послідовність дій, які виконуються під час реалізації конкретного варіанту використання або функціонування системи загалом [8].

**Діаграми діяльності (activity diagrams)** відображають послідовність дій, які виконуються в процесі реалізації, наприклад, для ідентифікації рослини. Вони, як і діаграми станів і переходів, представлені у вигляді орієнтованого графа, де вершини позначають дії, а ребра – переходи між ними.

**Діяльність (activity)** є специфічним випадком стану (state) без назви, яка має одну вхідну подію (OnEntry action). Кожна діяльність має назву, що складається з дієслова і кількох пояснюючих слів, таких як «Зробити фото рослини» або «Отримати результати ідентифікації» [9].

У діаграмах діяльності події (events) на переходах не визначаються, оскільки вважається, що перехід від однієї дії до іншої відбувається без умови. Гранична умова (guard condition) використовується для визначення дії, до якої переходить управління у випадку неоднозначності. Тобто, якщо з певної вершини на діаграмі

Програмне забезпечення ідентифікації рослин на основі методів штучного інтелекту діяльності можливий перехід до декількох інших вершин, для всіх переходів слід визначити граничну умову [10].

Характеристика дії (action) для переходу також не має значення, оскільки всі дії на діаграмі представлені вершинами графа.

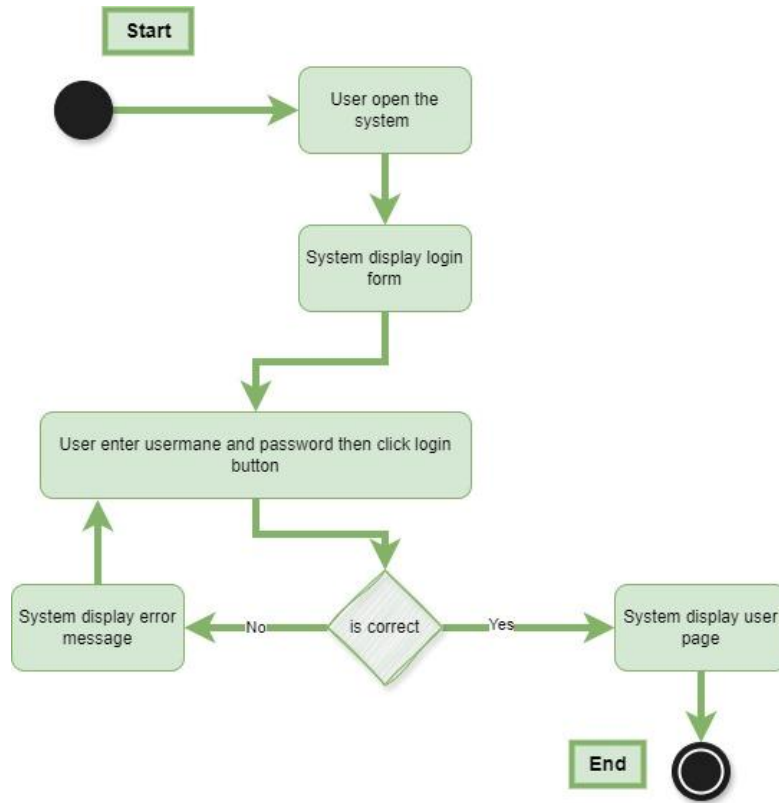


Рисунок 2.2 – Діаграма діяльності для входу в обліковий запис

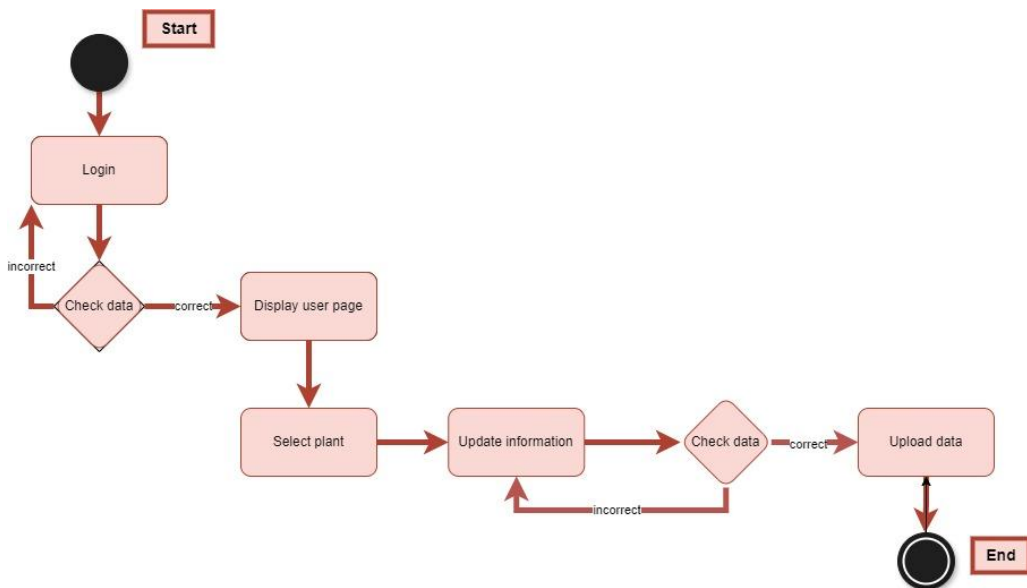


Рисунок 2.3 – Діаграма діяльності для оновлення даних про рослини

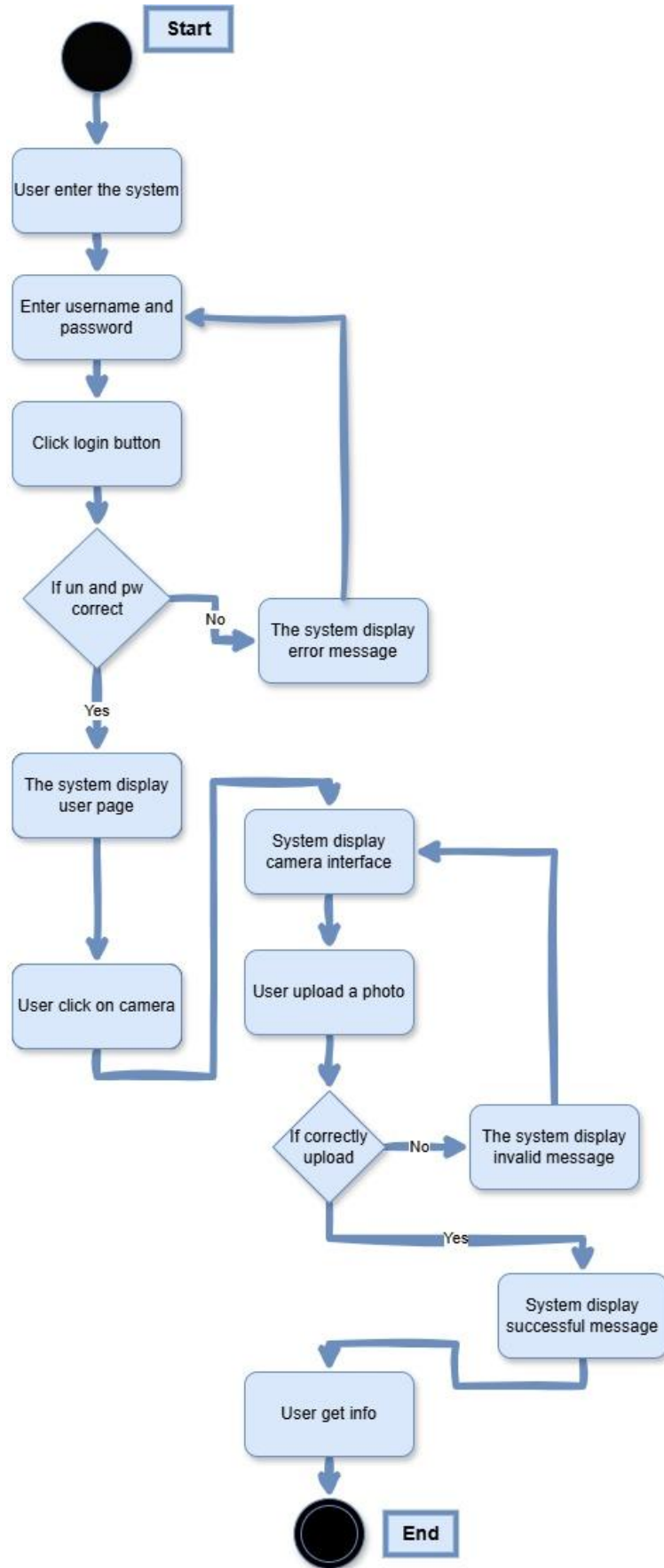


Рисунок 2.4 – Діаграми діяльності для отримання інформації про рослини

Діаграми послідовності надають чітке уявлення про динаміку взаємодії між об'єктами в системі (рис. 2.2–2.4). Вони допомагають візуалізувати порядок виконання операцій, що є важливим для розуміння логіки програми. Крім того, такі діаграми полегшують виявлення помилок і слабких місць у процесах. У результаті вони стають ключовим інструментом для розробки та вдосконалення програмного забезпечення.

## 2.4 Розробка діаграми розгортання

На рис. 2.5 представлена діаграма розгортання застосунку.

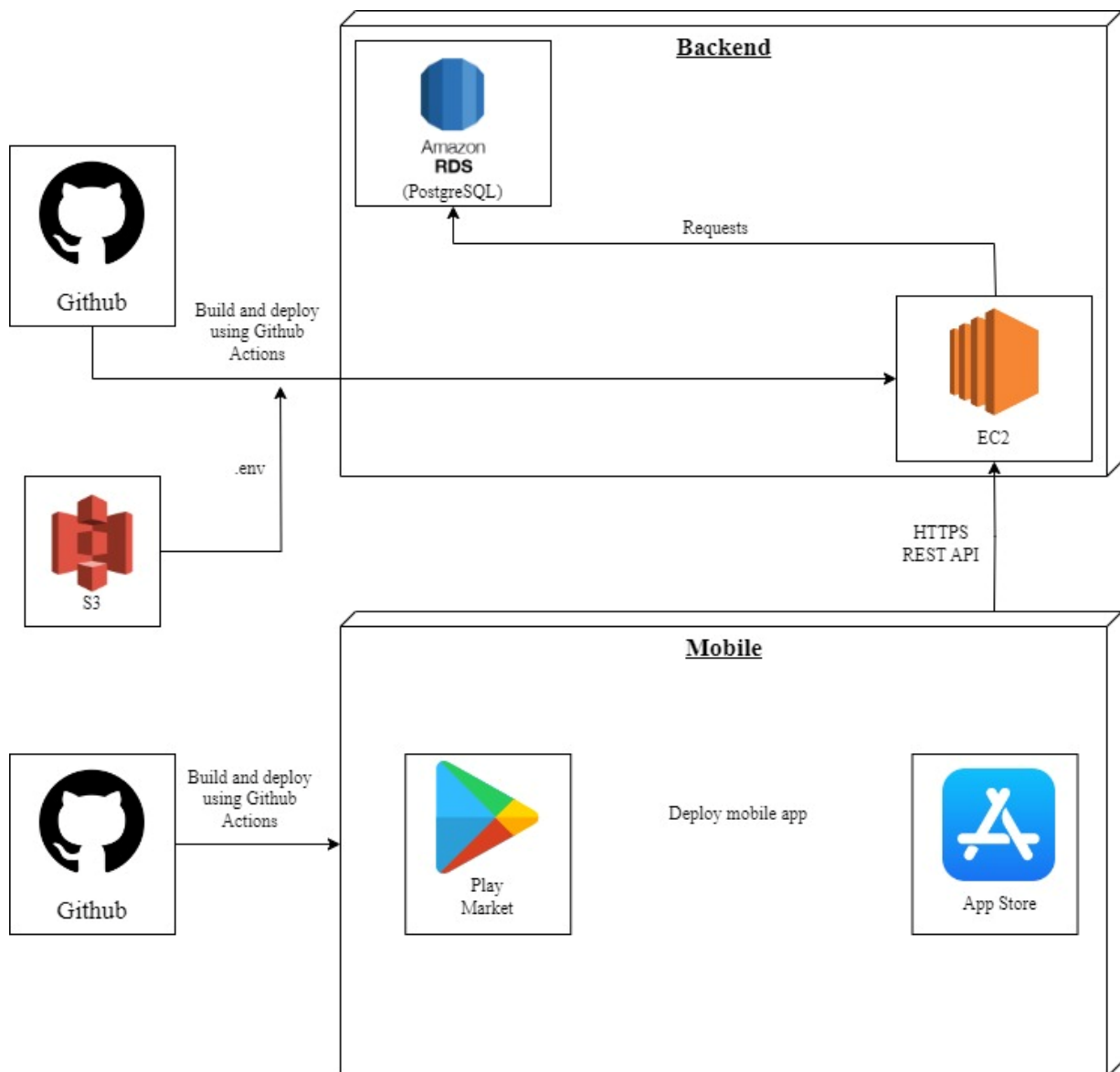


Рисунок 2.5 – Діаграма розгортання застосунку ідентифікації рослин

## **Опис компонентів діаграми розгортання**

### **AMAZON ELASTIC COMPUTE CLOUD (EC2)**

**Amazon EC2** – це потужна хмарна обчислювальна платформа, що пропонує понад 500 різноманітних інстансів. Вона підтримує найсучасніші процесори, сховища, мережеві рішення, операційні системи та варіанти покупки, щоб задовольнити будь-які робочі навантаження. Як один із провідних хмарних провайдерів, EC2 підтримує процесори Intel, AMD та Arm, і єдина платформа, яка дозволяє запускати macOS на інстансах Mac на вимогу, а також забезпечує мережеву пропускну здатність Ethernet до 400 Гбіт/с. EC2 вирізняється оптимальним співвідношенням ціни і продуктивності для проєктів машинного навчання та найнижчими витратами на запуск ML-інстансів у хмарі. AWS є лідером серед провайдерів для SAP, високопродуктивних обчислень (HPC), ML та Windows-середовищ [11].

### **Приклади використання**

- 1) Запуск корпоративних та хмарних застосунків.

Amazon EC2 забезпечує надійну, безпечну та масштабовану інфраструктуру для складних бізнес-застосунків.

- 2) Масштабування HPC-застосунків.

Швидкий доступ до необхідних ресурсів для економічного запуску програм високопродуктивних обчислень.

- 3) Розробка для Apple.

Можливість створювати, тестувати та випускати проєкти на macOS за лічені хвилини з динамічним масштабуванням ресурсів.

- 4) Навчання та розгортання ML-застосунків.

Amazon EC2 надає великий вибір рішень для обчислень, зберігання та мереж з оптимізованими витратами для завдань машинного навчання [12].

### **AMAZON RELATIONAL DATABASE SERVICE (RDS)**

**Amazon Relational Database Service (RDS)** – це зручний набір керованих інструментів, що спрощує процеси налаштування, управління та масштабування

баз даних у хмарі. Користувачі можуть обрати серед семи популярних платформ: Amazon Aurora, сумісна з MySQL чи PostgreSQL, а також MySQL, MariaDB, PostgreSQL, Oracle та SQL Server. Завдяки AWS Outposts, ці бази можна розгорнути навіть у локальному середовищі [13].

### Приклади використання

- 1) Розробка мобільних та вебзастосунків.

Забезпечте стабільну роботу застосунків із високою пропускнуою здатністю та можливістю масштабування. Гнучка модель оплати за фактом використання підходить для різних сценаріїв.

- 2) Перехід на керовані бази даних.

Зосередьтеся на інноваціях та розробці нових програм, мінімізуючи витрати на управління базами даних завдяки Amazon RDS.

- 3) Міграція зі старих баз даних.

Уникайте витрат і складнощів з ліцензуванням комерційних баз даних, перейшовши на Amazon Aurora, яка надає продуктивність і доступність на рівні комерційних систем, але у десять разів дешевше [14].

### AMAZON SIMPLE STORAGE SERVICE (S3)

**Amazon S3** – це хмарний сервіс зберігання об'єктів, який забезпечує високу продуктивність, масштабованість, доступність та безпеку даних. Незалежно від розміру компанії чи галузі, користувачі можуть надійно зберігати та захищати великі обсяги даних для будь-яких потреб, включаючи озера даних, хмарні чи мобільні застосунки. Зручні класи зберігання та прості інструменти управління дозволяють ефективно контролювати витрати, упорядковувати дані та налаштовувати доступ відповідно до вимог бізнесу та законодавства [15].

### Приклади використання

- 1) Створення озер даних.

Використовуйте Amazon S3 для запуску аналітики великих даних, штучного інтелекту (AI), машинного навчання (ML) та високопродуктивних обчислень (HPC).



## 2) Резервне копіювання та відновлення даних.

Виконуйте нормативи щодо відновлення часу (RTO), точок відновлення (RPO) та законодавчих вимог за допомогою можливостей реплікації Amazon S3 [16].

## 3) Архівація даних за низькою ціною.

Зменшуйте витрати на зберігання архівів за допомогою класів S3 Glacier, позбавляючи себе складних операційних процесів та відкриваючи нові можливості для аналізу.

## 4) Запуск хмарних програм.

Розробляйте швидкі, масштабовані мобільні та хмарні застосунки, які автоматично адаптуються до високих навантажень та забезпечують стабільну роботу.

## **GITHUB ACTIONS**

**GitHub Actions** – це платформа для автоматизації безперервної інтеграції та доставки (CI/CD), яка допомагає автоматизувати процеси збірки, тестування та розгортання. Ви можете створювати робочі процеси для автоматичного тестування кожного pull request або для розгортання змін у робоче середовище після їх об'єднання в основну гілку.

GitHub Actions не обмежується лише DevOps. Ви можете налаштувати робочі процеси для запуску при будь-яких подіях у вашому репозиторії. Наприклад, автоматично додавати теги до нових задач або реагувати на інші дії [17].

Платформа надає віртуальні машини на основі Linux, Windows і macOS для виконання ваших робочих процесів, або ж ви можете розміщувати власні сервери в хмарі чи на своїх дата-центрах.

## **GOOGLE PLAY TA APP STORE**

**Google Play і App Store** – це платформи для розповсюдження мобільних застосунків, відповідно, для Android та iOS. Вони дозволяють розробникам публікувати та оновлювати мобільні застосунки, забезпечуючи доступ користувачам до нових версій. Після завантаження застосунків на ці платформи,

вони стають доступними для завантаження, встановлення та автоматичного оновлення на мобільних пристроях.

## 2.5 Побудова та використання діаграм послідовності

На рис. 2.6 показана діаграма послідовності для основного сценарію використання «Вхід у систему» [18]. Під рисунком наводиться короткий опис цієї діаграми.

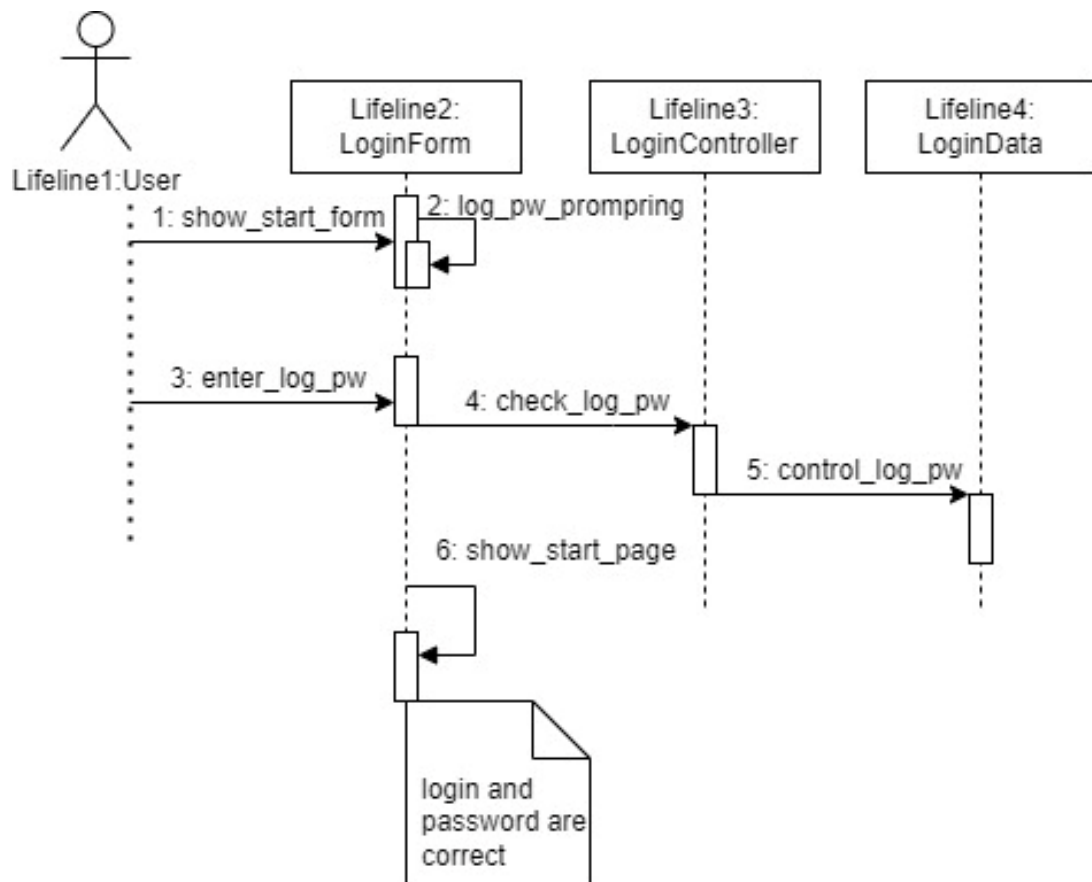


Рисунок 2.6 – Діаграма послідовності для основного потоку подій варіанту використання «Вхід у систему»

Цей варіант використання починається з того, що користувач запускає застосунок. Після цього відкривається форма, де потрібно ввести логін і пароль. Після отримання введених даних система виконує їх перевірку. Якщо дані підтверджено, система призначає користувачу відповідний рівень доступу залежно від його ролі і відображає необхідну інформацію на екрані.

На рисунку 2.7 показана діаграма послідовності для додаткового сценарію використання «Ідентифікація рослини» [19].

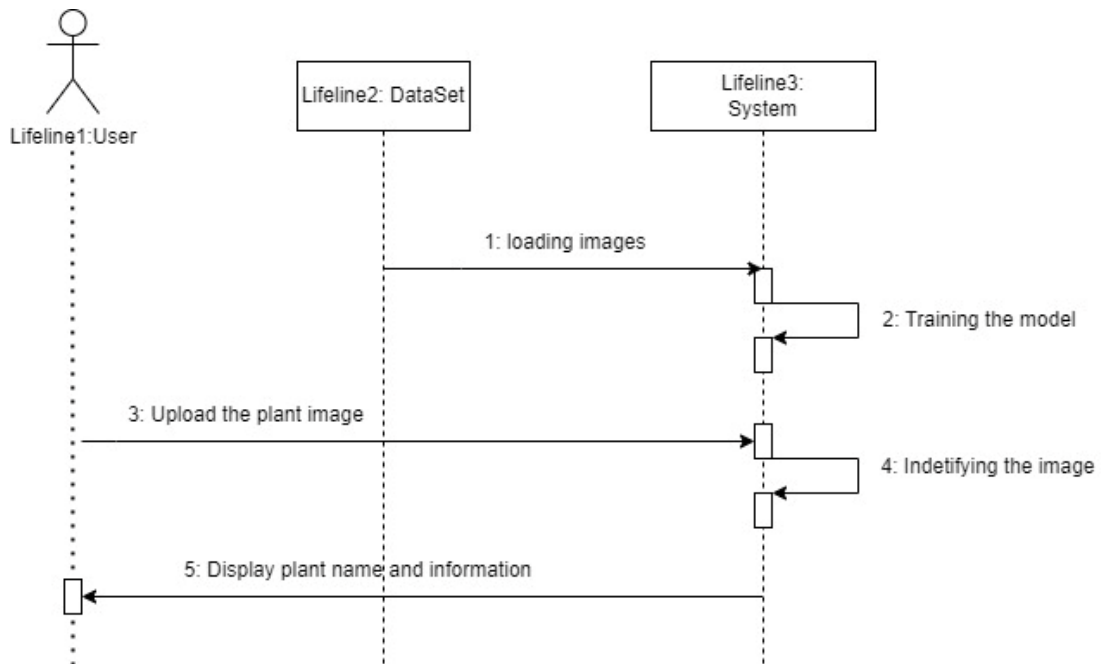


Рисунок 2.7 – Діаграма послідовності для додаткового потоку подій варіанту використання «Ідентифікація рослини»

Ініціатором даного варіанту використання є користувач. Після авторизації в системі, користувачу відкривається можливість завантажити фото рослини. Система звертається до набору даних для попереднього завантаження зображень та тренування моделі на основі цих даних. Після цього система використовує отримане зображення для ідентифікації рослини і, якщо процес ідентифікації успішний, система повертає користувачу інформацію про рослину, включаючи її назву.

На рисунку 2.8 показана діаграма послідовності для основного потоку подій варіанту використання «Доступ до управління інформацією про рослини».

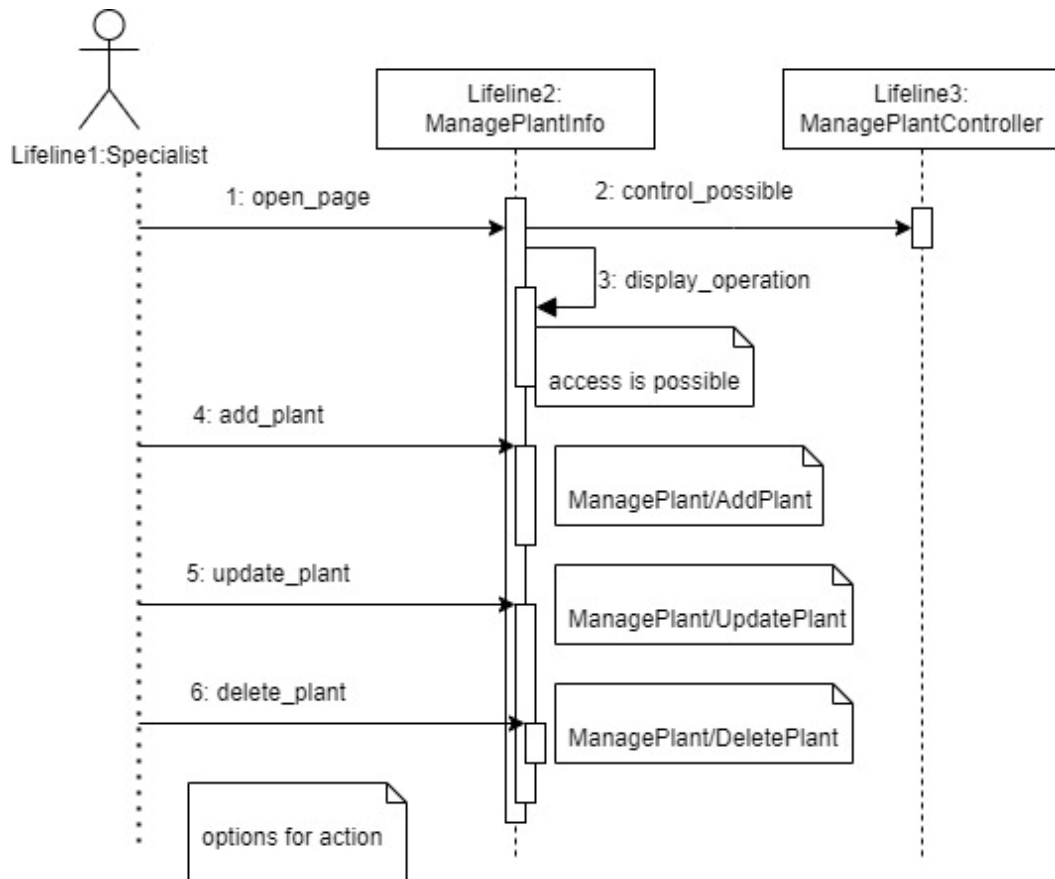


Рисунок 2.8 – Діаграма послідовності для основного потоку подій варіанту використання «Доступ до управління інформацією про рослини»

Ініціатором даного варіанту використання є науковий співробітник/фахівець. Після успішної авторизації в системі відкривається форма, яка містить перелік доступних дій, тому фахівець може обрати необхідну дію з запропонованих варіантів. Після отримання вибору фахівця, система відкриває відповідну форму.

На рис. 2.9 представлений варіант використання, який демонструє послідовність дій між основними компонентами системи ідентифікації рослин.

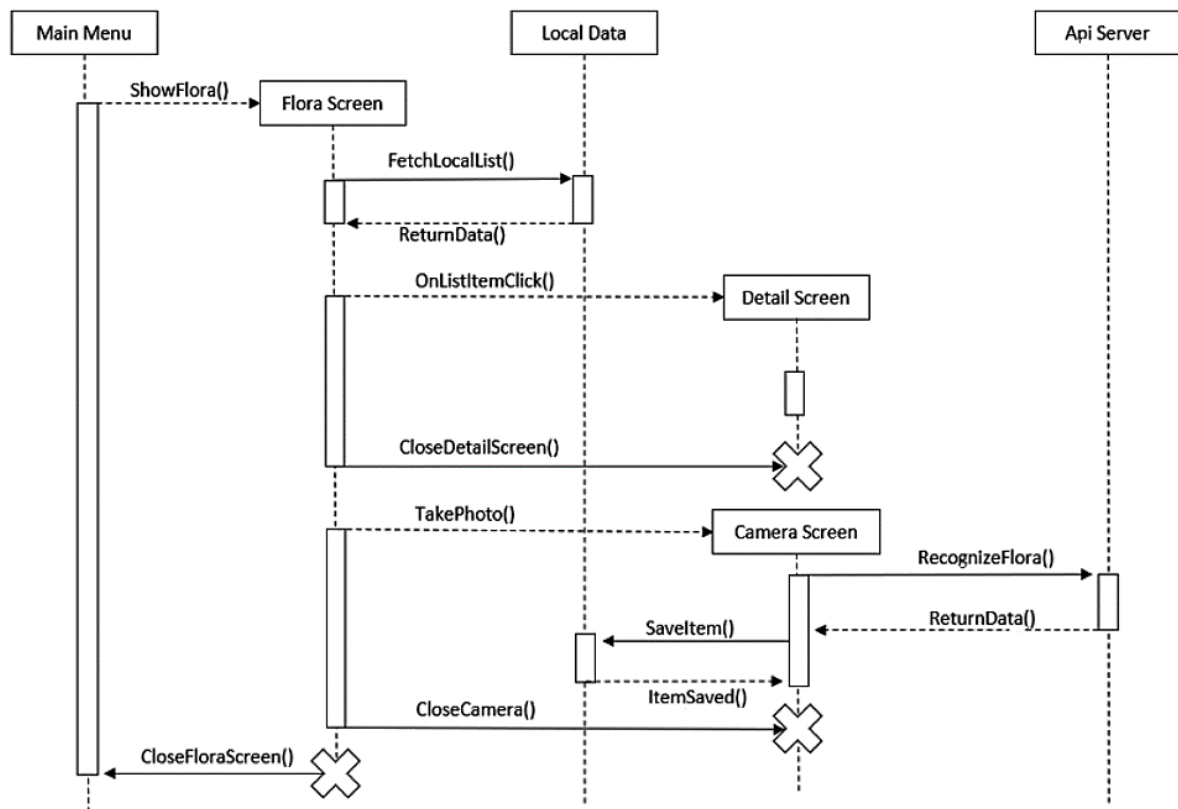


Рисунок 2.9 – Діаграма послідовності для додаткового потоку подій варіанту використання «Ідентифікація рослини»

Даний варіант використання описує загальний процес взаємодії користувача із системою ідентифікації рослин. Спочатку відображаються дані з локальної бази, які користувач може переглядати. Натискання на певний елемент відкриває детальний екран для отримання додаткової інформації. Далі користувач може зробити фото рослини, яке зберігається локально та надсилається на сервер для аналізу. Результати ідентифікації повертаються системою, після чого користувач може завершити роботу та повернутися до головного меню. У процесі забезпечується перехід між екранами та взаємодія з локальною і віддаленою базами даних.

## **Висновки до розділу 2**

У другому розділі проведено моделювання застосунку для ідентифікації рослин на основі методів штучного інтелекту. Розглянуто ключові етапи реалізації проекту, що забезпечують чітке розуміння процесу впровадження застосунку.

Створено різні варіанти використання (USE CASE), які описують основні функції системи та взаємодію користувачів з нею. Розписано короткі, поверхневі та альтернативні сценарії використання, що допомагає визначити можливі варіанти дій у різних ситуаціях.

Описано функції та ролі всіх дійових осіб, що взаємодіють із системою, включаючи користувачів, спеціалістів і адміністрацію, що забезпечує зрозумілу структуру роботи. Розроблено алгоритм роботи програмного забезпечення, а також діаграми діяльності, які ілюструють послідовність дій користувачів під час роботи з застосунком.

Побудовано та описано діаграми послідовності, що надають змогу візуалізувати взаємодію між компонентами системи, що сприяє глибшому розумінню її функціонування. Весь представлений матеріал створює міцну основу для подальшої реалізації мобільного застосунку для ідентифікації рослин на основі методів штучного інтелекту.

### 3 ПРОЄКТУВАННЯ СИСТЕМИ ІДЕНТИФІКАЦІЇ РОСЛИН

#### 3.1 Архітектура системи

Програмне забезпечення для ідентифікації рослин, що використовує методи штучного інтелекту, побудоване за клієнт-серверною архітектурою (рис. 3.1). Воно складається з серверного компоненту, реалізованого на базі Nest.js, Python для роботи з моделями машинного навчання, бази даних Postgres, а також клієнтських частин: мобільного застосунку на React Native та адміністративної панелі, створеної на React Admin.

Ця архітектура забезпечує низку важливих переваг:

- *технологічна гнучкість*: поєднання Python для виконання завдань машинного навчання, Nest.js для управління серверними процесами та React Native для розробки мобільного інтерфейсу дозволяє адаптувати систему під специфічні потреби;

- *масштабування компонентів*: архітектура дозволяє незалежно збільшувати продуктивність кожного модуля, що особливо важливо для обробки великого обсягу зображень або запитів користувачів;

- *висока надійність*: завдяки поділу обов'язків між компонентами зменшується ризик критичних збоїв у роботі всієї системи;

- *розподіл функціональних завдань*: Python виконує обчислення, пов'язані з розпізнаванням рослин та обробкою даних, тоді як Nest.js відповідає за управління API і забезпечує зв'язок між клієнтами та сервером;

- *легкість інтеграції*: використання стандартних протоколів і REST API дає змогу легко підключати систему до зовнішніх сервісів і розширювати її функціонал;

- *продуктивність і оптимізація*: спеціалізація кожного компонента на конкретних завданнях дозволяє досягти максимальної ефективності, наприклад, Nest.js для обробки запитів у реальному часі, а Python для складних обчислень моделей штучного інтелекту;

– *посилена безпека*: модульна архітектура сприяє ізоляції важливих процесів, таких як зберігання та аналіз даних, що підвищує захист системи від потенційних загроз.

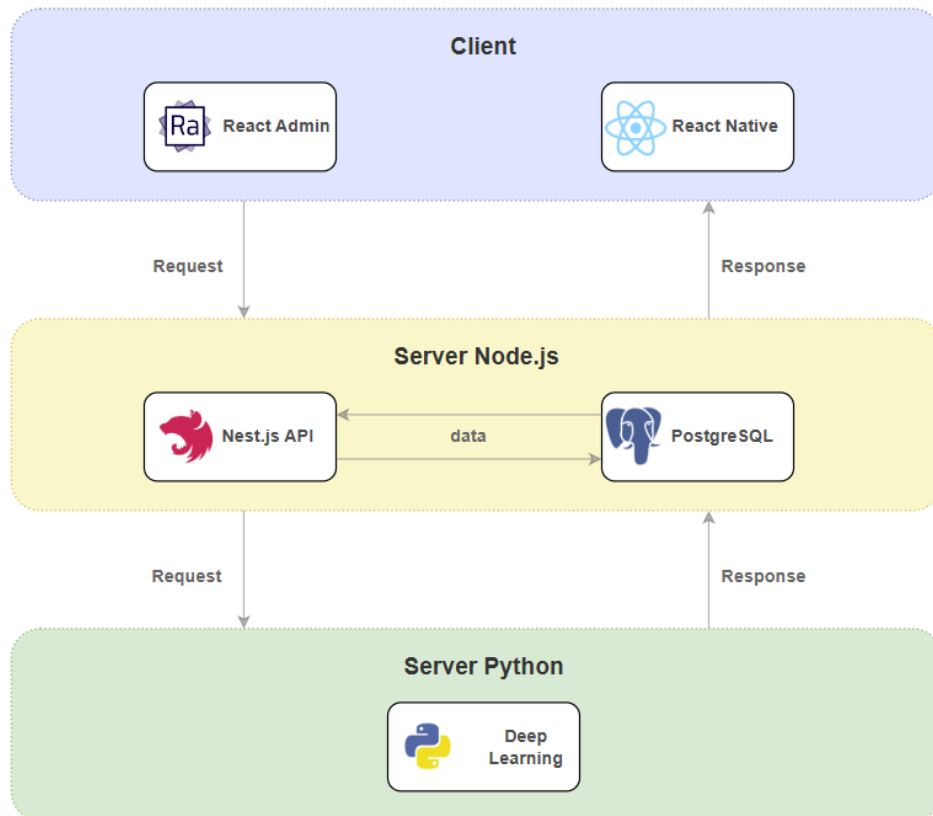


Рисунок 3.1 – Архітектура клієнт-сервер

Реалізована архітектура пропонує універсальне та продуктивне рішення для створення складних інтелектуальних систем, здатних забезпечити високу ефективність, стійкість і можливість масштабування.

### 3.2 Структура бази даних

Для створення програмного забезпечення ідентифікації рослин було детально розроблено структуру бази даних для збереження різнопланових сутностей з можливістю масштабування в майбутньому.

Для успішної реєстрації та ідентифікації користувачів було додано таблицю “users” (рис. 3.2).





user	
123	id
A-Z	email
A-Z	password
A-Z	phone
A-Z	googleId
A-Z	username
🕒	createdAt
🕒	updatedAt
123	roleId
123	settingsId
123	profileId

Рисунок 3.2 – Таблиця “users”

Таблиця “users” має наступну структуру:

- *id*: унікальний ідентифікатор користувача;
- *email*: пошта;
- *password*: пароль;
- *phone*: номер телефону;
- *googleId*: ідентифікатор акаунту Google;
- *username*: нікнейм користувача;
- *roleId*: ідентифікатор ролі;
- *settingsId*: ідентифікатор налаштувань користувача;
- *profileId*: ідентифікатор профілю.

При реєстрації створюється новий запис в таблиці users, в якій додаються ті поля, які користувач ввів при заповненні форми. Щоразу при логіні в застосунок перевіряється введений пароль та пошта на відповідність акаунту, а також перевіряється ідентифікатор Google, якщо користувач обрав можливість авторизації за допомогою GoogleAuth.

Далі створено таблицю “role” (рис. 3.3), яка використовується для визначення доступу користувача до певних функцій застосунку.



Рисунок 3.3 – Таблиця “role”

Таблиця “roles” має наступні поля:

- *id*: унікальний ідентифікатор;
- *name*: назва ролі;

Таблиця відповідає за доступність функціоналу перед користувачем. Наразі створено 3 сутності – user, admin, expert. Адміністратор має можливість використовувати повний спектр можливостей застосунку, таких як перегляд адміністративної панелі та додавання нових рослин до бази даних. Науковий фахівець, він же expert, має змогу відповідати на запит про допомогу, а також створювати запит оновлення певних даних стосовно рослин у разі виявлення помилки. Звичайний користувач має право на перегляд рослин та їх ідентифікацію.

Далі створено таблицю “profile” (рис. 3.4), що включає в себе важливий зв’язок з таблицею users, доповнюючи її важливої інформацією.



Рисунок 3.4 – Таблиця “profile”

До таблиці “profile” входять поля:

- *id*: унікальний ідентифікатор;

- *fullName*: повне ім'я;
- *gender*: стать людини;
- *birthday*: дата народження;
- *avatar*: посилання на фото профілю.

Далі створено таблицю “settings” (рис. 3.5), яка допомагає користувачу зберігати свої налаштування, що буде корисним при використанні одного акаунту на декількох пристроях.

settings	
123	id
	createdAt
	updatedAt
<input checked="" type="checkbox"/>	notificationsOn
<input checked="" type="checkbox"/>	accountUpdatesOn
<input checked="" type="checkbox"/>	contentRecommendationsOn

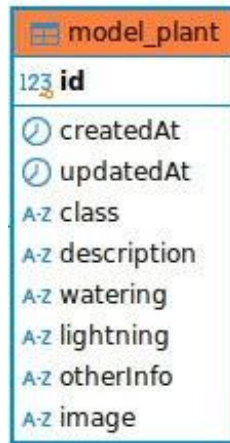
Рисунок 3.5 – Таблиця “settings”

У таблиці “settings” є поля:

- *id*: унікальний ідентифікатор;
- *notificationsOn*: стан налаштування повідомлень;
- *accountUpdatesOn*: стан налаштування оновлень для типу акаунту;
- *contentRecommendationsOn*: стан налаштування рекомендацій контенту.

Налаштування є важливим доповненням застосунку, що дозволяє універсально синхронізувати акаунт через використання різних девайсів.

На рис. 3.6 представлено таблицю “model\_plant”, що відповідає за зберігання ідентифікованих рослин.



model_plant	
123	id
	createdAt
	updatedAt
	A-Z class
	A-Z description
	A-Z watering
	A-Z lightning
	A-Z otherInfo
	A-Z image

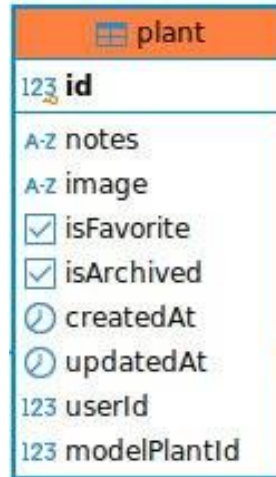
Рисунок 3.6 – Таблиця “model\_plant”

Серед основних полів таблиці є:

- *id*: унікальний ідентифікатор;
- *class*: класифікація рослини;
- *description*: опис;
- *watering*: частота поливу рослини;
- *lightning*: необхідність освітлення для рослини;
- *otherInfo*: додаткова корисна інформація;
- *image*: фото, що презентує рослину.

Таблиця “model\_plant” є однією з найважливіших таблиць у системі, так як відповідає за надання інформації про рослини після ідентифікації. Дана таблиця наповнюється даними при тренуванні моделі з файлу metadata.csv, що входить у датасет.

Таблиця “plant” (рис. 3.7) є важливим компонентом, що використовується при ідентифікації рослини, так як тут зберігаються всі рослини користувача.



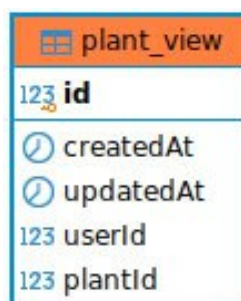
plant	
123	id
A-Z	notes
A-Z	image
<input checked="" type="checkbox"/>	isFavorite
<input checked="" type="checkbox"/>	isArchived
🕒	createdAt
🕒	updatedAt
123	userId
123	modelPlantId

Рисунок 3.7 – Таблиця “plant”

Для використання таблиці “plant” додано наступні поля:

- *id*: унікальний ідентифікатор;
- *notes*: замітки про рослину;
- *image*: фото рослини;
- *isFavorite*: позначка улюблених рослин;
- *isArchived*: позначка архівованих рослин;
- *userId*: ідентифікатор користувача;
- *modelPlantId*: ідентифікатор шаблону рослини.

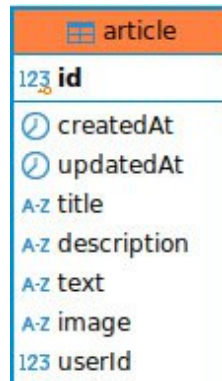
Для відстеження статистики переглядів рослин була створена таблиця “plant\_view” (рис. 3.8), що посилається на конкретну рослину та користувача, і містить ті ж основні поля, необхідні для ідентифікації перегляду.



plant_view	
123	id
🕒	createdAt
🕒	updatedAt
123	userId
123	plantId

Рисунок 3.8 – Структура таблиці “plant\_view”

Таблиця “article” (рис. 3.9) стала основним інструментом для зберігання та управління інформацією про публікації в системі. Вона дозволяє зберігати дані про назву, опис, текст статті та пов’язане з нею зображення. Ця таблиця також прив’язана до конкретного користувача, що забезпечує можливість фільтрування та персоналізації контенту.



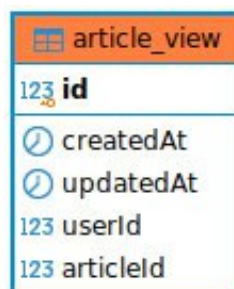
article	
123	id
	createdAt
	updatedAt
A-Z	title
A-Z	description
A-Z	text
A-Z	image
123	userId

Рисунок 3.9 – Структура таблиці “article”

Полями таблиці є:

- *id*: унікальний ідентифікатор;
- *title*: назва статті;
- *description*: короткий опис статті;
- *text*: основний текст статті;
- *image*: зображення, пов’язане зі статтею;
- *userId*: ідентифікатор користувача.

Таблиця “article\_view” (рис. 3.10) забезпечує фіксацію інформації про те, які статті переглядалися користувачами. Ця таблиця дозволяє відстежувати популярність контенту та взаємодію користувачів із публікаціями.



article_view	
123	id
	createdAt
	updatedAt
123	userId
123	articleId

Рисунок 3.10 – Структура таблиці “article\_view”

Полями таблиці є:

- *id*: унікальний ідентифікатор;
- *userId*: ідентифікатор користувача;
- *articleId*: ідентифікатор статті.

Таблиця “favorite\_article” (рис. 3.11) дозволяє зберігати інформацію про статті, які користувачі додали до списку обраного. Вона використовується для персоналізації контенту та забезпечення швидкого доступу до улюблених публікацій.



favorite_article	
123	id
	createdAt
	updatedAt
123	userId
123	articleId

Рисунок 3.11 – Структура таблиці “favorite\_article”

Полями таблиці є:

- *id*: унікальний ідентифікатор;
- *userId*: ідентифікатор користувача;
- *articleId*: ідентифікатор статті.

Таблиця “notification” (рис. 3.12) використовується для зберігання інформації про повідомлення, які надсилаються користувачам. Це дозволяє ефективно відслідковувати важливі події та інформувати користувачів про новини чи оновлення.

notification	
123	id
	⌚ createdAt
	⌚ updatedAt
A-Z	title
A-Z	description
A-Z	icon
<input checked="" type="checkbox"/>	isRead
123	userId

Рисунок 3.12 – Структура таблиці “notification”

Полями таблиці “notification” є:

- *id*: унікальний ідентифікатор;
- *title*: заголовок повідомлення;
- *description*: детальний опис повідомлення;
- *icon*: іконка, яка відображається поряд з повідомленням;
- *isRead*: статус перегляду повідомлення;
- *userId*: ідентифікатор користувача.

Таблиця “faq” (рис. 3.13) містить питання та відповіді, що надаються користувачам для швидкого отримання інформації.

faq	
123	id
	⌚ createdAt
	⌚ updatedAt
A-Z	question
A-Z	answer

Рисунок 3.13 – Структура таблиці “faq”

Полями таблиці “faq” є:

- *id*: унікальний ідентифікатор;
- *question*: текст запитання;
- *answer*: текст відповіді на запитання.



Таблиця “request” (рис. 3.14) зберігає інформацію про запити користувачів до експертів, а також оновлення тексту про рослину, яке надається експертом.

request	
123	id
🕒	createdAt
🕒	updatedAt
A-Z	text
A-Z	type
A-Z	status
123	authorId

Рисунок 3.14 – Структура таблиці “request”

Полями таблиці “request” є:

- *id*: унікальний ідентифікатор;
- *text*: текст запиту або оновленої інформації про рослину;
- *type*: тип запиту;
- *status*: статус запиту;
- *authorId*: ідентифікатор автора запиту.

Загальна структура бази даних (рис. 3.15) була розроблена для забезпечення ефективного зберігання та обробки даних, що стосуються користувачів, статей, рослин, запитів та повідомлень у системі. Вона включає таблиці, які мають чіткі зв'язки між собою через первинні та зовнішні ключі, що дозволяє забезпечити цілісність даних і уникнути їх редундантності.

Таблиці, такі як “article”, “user”, “plant”, “request”, “notification”, та інші, взаємодіють через зовнішні ключі для збереження зв'язків між різними сутностями, такими як користувачі, статті, перегляди, улюблені статті та запити до експертів.

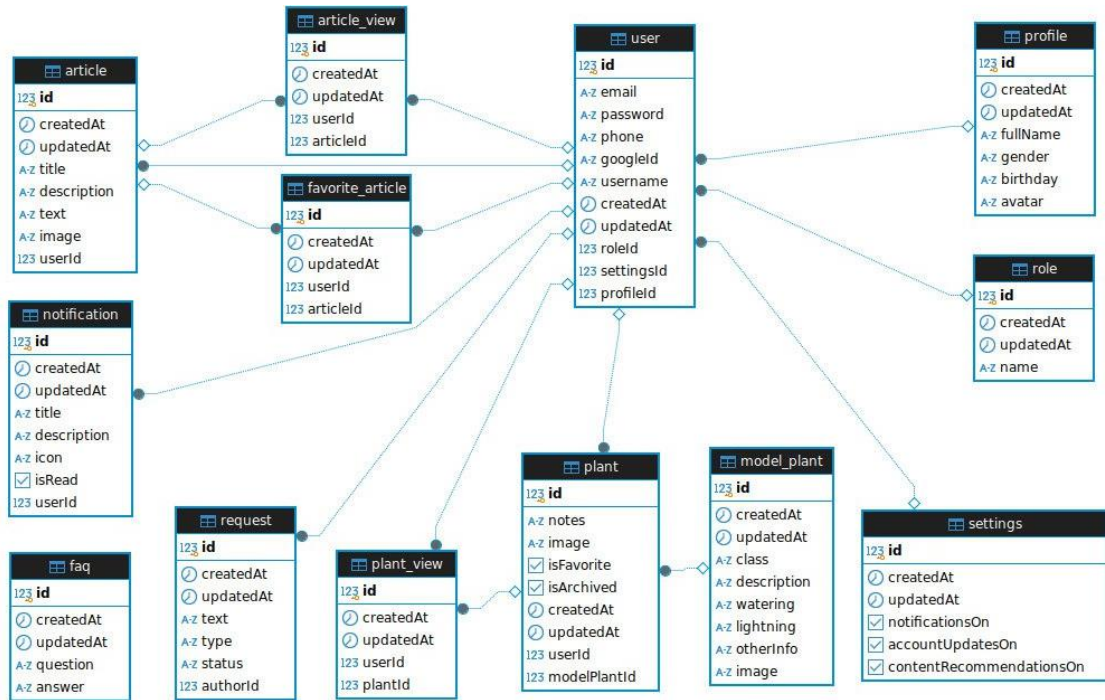


Рисунок 3.15 – ER-діаграма бази даних

Крім того, забезпечено високу нормалізацію бази даних для покращення масштабованості та ефективності роботи системи, що дозволяє легко обслуговувати та оновлювати дані.

Кожна таблиця містить важливі поля для зберігання ключової інформації, наприклад, у таблиці “article” зберігаються поля, що відносяться до статей (назва, текст, автор), у таблиці “request” зберігається інформація про запити користувачів до експертів, а в таблиці plant зберігаються дані про рослини.

Така структура дозволяє легко відстежувати взаємодії користувачів з контентом, а також зберігати історію запитів та переглядів статей і рослин.

### 3.3 Розробка датасету

Для забезпечення ефективного тренування моделі класифікації рослин створено і структуровано спеціалізований датасет, який включає зображення різних видів рослин.

Основним джерелом для формування датасету були відкриті ресурси з зображеннями рослин, зокрема набори даних, створені для задач комп’ютерного

зору, та онлайн-бібліотеки зображень. Під час збору увага приділялася різноманітності даних: до датасету включалися зображення рослин, зроблені під різними кутами, у різному освітленні та на різному фоні. Це дозволило покращити здатність моделі розпізнавати рослини у реальних умовах.

Зображення були структуровані у форматі директорій, де кожна папка відповідала певному класу рослин. Наприклад, зображення ромашок зберігалися в директорії Daisy, троянд — у директорії Rose, а соняшників — у Sunflower. Така організація спрощує обробку та завантаження даних у моделі глибокого навчання.

Після збору датасету було виконано попередню обробку зображень, щоб зробити їх придатними для використання в моделі глибокого навчання. Основні етапи обробки включали:

- масштабування зображень було змінено до фіксованого розміру. Це стандартний розмір для моделей на основі EfficientNet, який дозволяє забезпечити однаковий формат вхідних даних;
- кожне зображення було нормалізоване відповідно до середнього та стандартного відхилення, характерного для датасету ImageNet, на якому попередньо навчалася архітектура EfficientNet. Це дозволило зменшити розбіжності у розподілі кольорів між різними зображеннями, що позитивно вплинуло на стабільність та точність тренування.

Окрім основних зображень, було створено таблицю метаданих у форматі CSV, яка містила описову інформацію про кожен клас рослин. Ця таблиця включає такі атрибути, як:

- назва рослини;
- короткий опис зовнішнього вигляду;
- рекомендації щодо догляду (наприклад, частота поливу, освітлення тощо).

Метадані інтегрувалися у кінцеву систему класифікації, що дало змогу не лише визначати клас рослини, але й надавати користувачеві корисну додаткову інформацію.

Підсумковий датасет складався з різних категорій рослин. Для кожної категорії було зібрано збалансовану кількість зображень, що забезпечило рівномірне представлення кожного класу в навчальних даних. Загальний обсяг датасету дозволив ефективно навчати модель, запобігаючи перекоосу в сторону одного з класів.

Для тестування моделі частина датасету була виділена в окремий тестовий набір (рис 3.16). Це дозволило об'єктивно оцінити точність і здатність моделі узагальнювати знання на нових, невідомих даних.

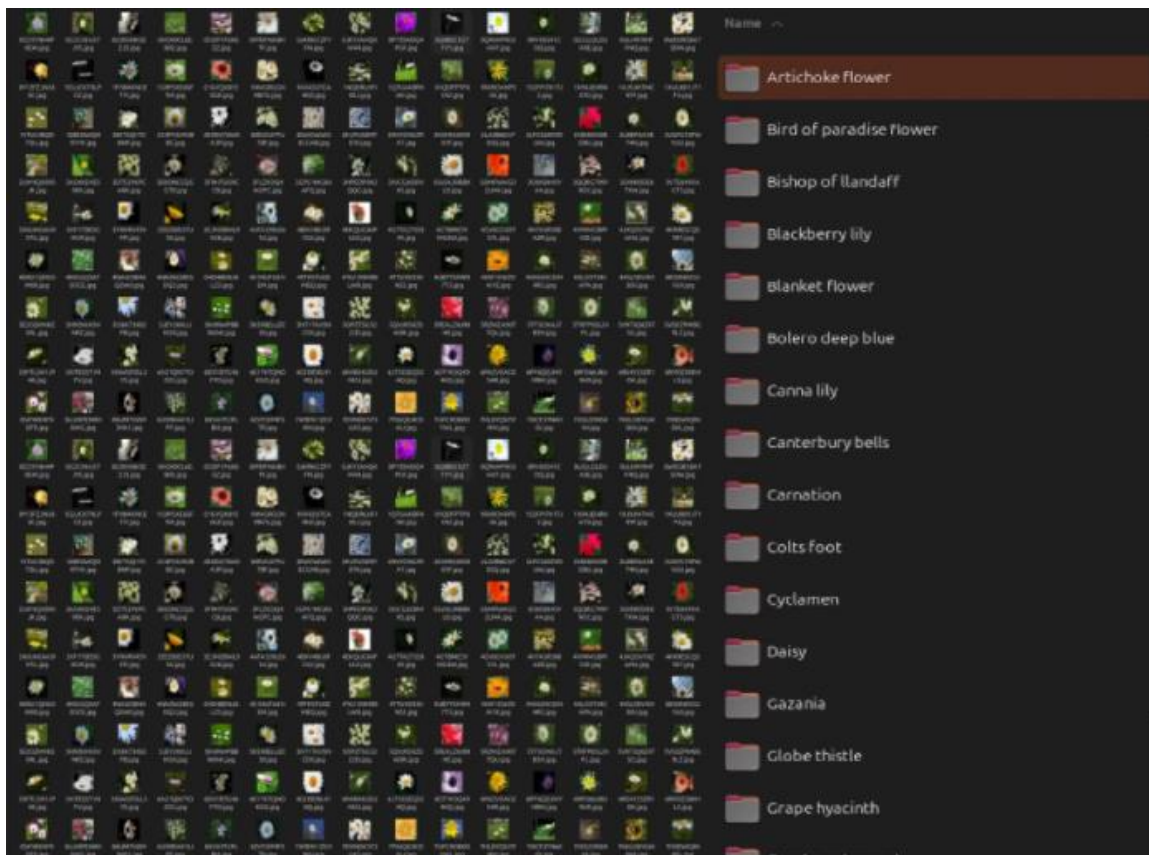


Рисунок 3.16 – Створений датасет

Підготовлений датасет відповідає всім вимогам для ефективного тренування моделі класифікації. Крім того, метадані розширюють функціональні можливості системи, роблячи її більш інформативною для кінцевого користувача. Цей датасет забезпечив високу якість моделі та дозволив досягти точних результатів при класифікації зображень рослин.

### **Висновки до розділу 3**

У третьому розділі розроблено ключові компоненти системи ідентифікації рослин, включаючи архітектуру, базу даних і датасет для навчання моделі машинного навчання.

Було визначено клієнт-серверну архітектуру системи, яка поєднує серверні технології Nest.js для управління API, Python для обробки даних моделі машинного навчання та мобільний застосунок на React Native. Такий підхід забезпечує модульність, масштабованість і надійність системи, дозволяючи інтегрувати нові функції та підтримувати стабільну роботу за умов зростання навантаження.

У процесі проєктування бази даних була створена структура, що враховує потреби системи в обробці, зберіганні та аналізі великих обсягів даних. Запропонована схема бази даних, оптимізована для швидкого доступу до інформації, забезпечує підтримку зберігання користувачів, їхніх ролей, даних про рослини та історії взаємодії із застосунком. Зв'язки між сутностями були чітко визначені, що сприяє збереженню цілісності даних.

Окрему увагу приділено розробці датасету для навчання та тестування моделі класифікації рослин. Було зібрано збалансований набір зображень різних видів рослин, структурований у форматі директорій за класами. Після попередньої обробки зображень, включаючи нормалізацію та масштабування, було створено високоякісний датасет, який дозволяє досягти високої точності при класифікації. Інтеграція метаданих забезпечує додаткову функціональність, розширюючи інформативність системи для кінцевих користувачів.

Таким чином, у розділі описано комплексний підхід до проєктування системи, що включає всі аспекти її реалізації: від архітектури до створення навчальних даних. Це дозволяє забезпечити ефективну роботу системи, її гнучкість та здатність адаптуватися до вимог користувачів і змінних умов експлуатації.

## 4 ПРОГРАМНА РЕАЛІЗАЦІЯ ЗАСТОСУНКА

### 4.1 Програмні засоби для розробки системи

Для реалізації поставлених задач, проведено ретельне дослідження щодо вибору найбільш ефективних інструментів розробки, які відповідали б вимогам системи для розпізнавання рослин за допомогою ШІ. З огляду на технічні вимоги, архітектурні особливості та наявні технології, для розробки було вибрано наступні програмні засоби:

- React Native;
- React Admin;
- Nest.js;
- Python;
- PostgreSQL;
- TensorFlow.

#### **React Native**

React Native – це популярний фреймворк для розробки мобільних застосунків, який дозволяє створювати застосунки для iOS та Android із єдиним кодом на JavaScript, використовуючи React. Його основна перевага – це кросплатформність, що дає змогу значно зменшити витрати на розробку, оскільки дозволяє працювати з однією базою коду для обох платформ.

Основні характеристики React Native:

- кросплатформність: можливість створення застосунків для iOS та Android без необхідності писати окремий код для кожної платформи;
- гнучкість і модульність: надає можливість інтеграції з нативними модулями та сторонніми бібліотеками для розширення функціональності;
- розширюваність: підтримка додавання власних компонентів або використання наявних бібліотек для зручного доступу до специфічних функцій пристроїв;

- **інтерактивність:** завдяки двосторонньому зв'язку даних між компонентами, зміни в одному компоненті миттєво відображаються в іншому, що забезпечує швидку реакцію на зміну стану;
- **широка спільнота та підтримка:** величезна кількість готових бібліотек і розширень, а також активна спільнота розробників, що дозволяє значно пришвидшити процес розробки.

React Native використовується для створення мобільного інтерфейсу користувача в системі, а також для інтеграції зі смартфоном для отримання зображень рослин для подальшого аналізу.

### **React Admin**

React Admin – це потужний фреймворк для створення адмін-панелей та інтерфейсів управління даними, що побудований на основі React. Він дозволяє швидко створювати складні адміністративні інтерфейси для управління даними, користувачами, ролями та іншими аспектами системи.

#### Основні характеристики React Admin:

- **простота інтеграції з API:** підтримка як REST, так і GraphQL API, що дає можливість легко інтегрувати React Admin з будь-яким серверним back-end;
- **гнучкий інтерфейс:** компонентний підхід забезпечує гнучкість у налаштуванні інтерфейсів для різних потреб користувачів;
- **управління даними:** надає готові рішення для роботи з таблицями, фільтрацією, сортуванням, створенням форм і перевіркою даних;
- **управління ролями та правами доступу:** можливість налаштовувати різні рівні доступу для адміністраторів і звичайних користувачів;
- **масштабованість:** система розроблена з урахуванням можливості обробки великої кількості даних та користувачів;
- **широка документація та підтримка:** надає просту і зрозумілу документацію, що спрощує швидкий старт розробки.

React Admin використовуватиметься для побудови інтерфейсу адмін-панелі, яка дозволить адміністраторам системи керувати користувачами, їх ролями, переглядати статистику використання та взаємодіяти з базою даних рослин.

### **Nest.js**

Nest.js – це прогресивний фреймворк для створення серверних застосунків на Node.js, заснований на TypeScript. Він використовує принципи об'єктно-орієнтованого програмування та декораторів, що дозволяє будувати чітко структуровані та масштабовані серверні рішення.

Основні характеристики Nest.js:

- модульна архітектура: підтримка модульного підходу, що полегшує організацію коду, повторне використання компонентів та зменшення залежностей між модулями;
- підтримка TypeScript: завдяки TypeScript зростає безпека коду, що зменшує ймовірність помилок;
- ін'єкція залежностей: зручна система управління залежностями, яка дозволяє ефективно працювати з великими проектами;
- підтримка REST та GraphQL: забезпечує гнучкість у виборі технологій для взаємодії між front-end і back-end;
- широка екосистема: можливість використання безлічі вбудованих бібліотек та пакетів для досягнення специфічних цілей.

Nest.js відповідає за створення серверної частини системи, яка обробляє запити від системи, взаємодіє з базою даних, обробляє зображення для подальшої передачі до моделі машинного навчання.

### **Python**

Python – це високорівнева мова програмування, що відзначається своєю простотою, потужною стандартною бібліотекою та підтримкою численних бібліотек для машинного навчання, обробки даних та штучного інтелекту. Python широко використовується в наукових обчисленнях і розробці систем, що потребують аналізу великих обсягів даних.



### Основні характеристики Python:

- читабельність та простота: простий синтаксис робить Python доступним для новачків і дозволяє швидко розробляти високоякісні застосунки;
- машинне навчання та штучний інтелект: завдяки популярним бібліотекам, таким як TensorFlow, Keras, PyTorch, Python є ідеальним вибором для розробки систем, що вимагають складних алгоритмів аналізу даних;
- широка екосистема: Python має великий набір бібліотек для обробки даних, зокрема Pandas, NumPy, Matplotlib;
- кросплатформність: Python підтримує всі основні операційні системи, що робить його зручним для кросплатформної розробки.

Python використовується для створення та тренування моделей машинного навчання, що відповідають за розпізнавання рослин за фотографіями.

### PostgreSQL

PostgreSQL – це потужна система управління реляційними базами даних, яка забезпечує надійність, масштабованість і високу продуктивність. PostgreSQL є однією з найкращих СУБД для обробки великих обсягів даних та підтримки складних запитів.

### Основні характеристики PostgreSQL:

- підтримка складних типів даних: можливість зберігання структурованих даних (JSON, XML), а також геопросторових даних;
- ACID-сумісність: підтримка транзакцій, що забезпечує консистентність та надійність роботи з базою даних;
- розширюваність: підтримка створення власних типів даних, функцій та індексів;
- масштабованість: висока продуктивність при роботі з великими обсягами даних.

PostgreSQL використовується для зберігання даних про рослини, користувачів, а також для зберігання результатів розпізнавання та звітів.

## **TensorFlow**

TensorFlow – це бібліотека з відкритим кодом для створення та тренування нейронних мереж, розроблена компанією Google. TensorFlow підтримує як глибоке навчання, так і інші методи машинного навчання, що робить її потужним інструментом для обробки зображень.

Основні характеристики TensorFlow:

- підтримка глибокого навчання: TensorFlow є однією з найкращих бібліотек для розробки та тренування глибоких нейронних мереж, таких як CNN (Convolutional Neural Networks) для обробки зображень;
- кросплатформність: підтримує запуск моделей на різних платформах, включаючи мобільні пристрої;
- масштабованість: можливість тренування моделей на великих наборах даних;
- гнучкість: TensorFlow дозволяє легко створювати різні архітектури моделей для різних типів задач.

TensorFlow використано для створення моделі, яка аналізує зображення рослин, визначає їх види та надає користувачам відповідну інформацію.

### **4.2 Опис програмної реалізації**

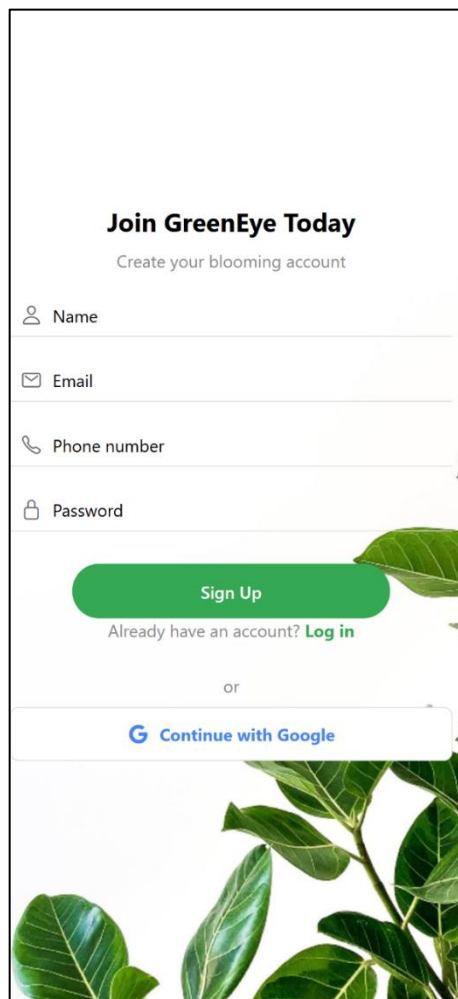
Програмна реалізація проекту включає створення комплексної системи, яка забезпечує розпізнавання рослин за зображеннями, а також управління інформацією про рослини та користувачів. Система складається з трьох основних компонентів: мобільного застосунку, серверної частини та адміністративної панелі. Архітектура побудована за принципами клієнт-серверної моделі. Система виконує роль клієнта, що надсилає запити до серверної частини, яка обробляє їх та взаємодіє з базою даних і моделлю машинного навчання. Адмінпанель використовується для управління даними та користувачами.

Мобільний застосунок, розроблений на React Native, забезпечує користувачам можливість:

- завантажувати зображення рослин для розпізнавання;
- переглядати інформацію про розпізнану рослину: її назву, особливості догляду, рекомендації щодо умов утримання тощо;
- керувати своїм акаунтом, включаючи зміну електронної пошти або пароля;
- використовувати локальне сховище даних для тимчасового збереження інформації про рослини до синхронізації з сервером.

Інтерфейс розроблений для забезпечення інтуїтивної навігації з мінімалістичним дизайном.

Реєстрація в застосунку (рис. 4.1) реалізована у вигляді зручної форми, яка дозволяє користувачам швидко створити обліковий запис. Інтерфейс виконаний у мінімалістичному стилі з акцентом на зручність і доступність.



**Join GreenEye Today**  
Create your blooming account

Name

Email

Phone number

Password

**Sign Up**

Already have an account? **Log in**

or

**Continue with Google**

Рисунок 4.1 – Сторінка реєстрації

Форма реєстрації візуально доповнена елементами дизайну, такими як зображення зелених рослин у нижній частині, які підкреслюють тематику застосунку.

Сторінка входу в застосунок (рис. 4.2) забезпечує користувачам зручний і зрозумілий спосіб авторизації. Інтерфейс оформлений у мінімалістичному стилі з яскравими акцентами, що відповідає загальному дизайну застосунка.

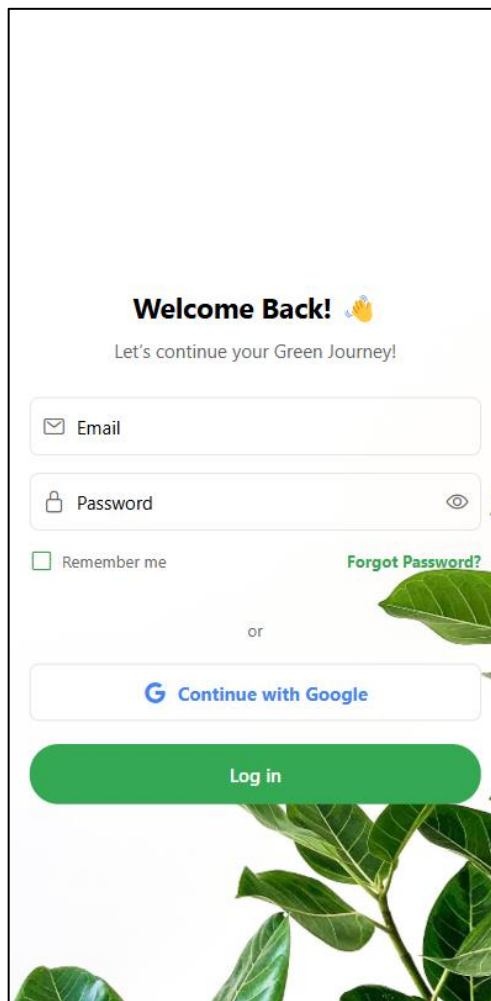


Рисунок 4.2 – Сторінка авторизації

Екран входу складається з декількох основних компонентів. Верхню частину займає великий заголовок, під яким розташовані два текстові поля для введення електронної пошти та пароля. Кожне з цих полів має відповідний підпис. Нижче цих полів є кнопка призначена для входу в систему. Під цією кнопкою розташована кнопка для входу через Google.

На самій нижній частині екрану є гіперпосилання для відновлення пароля. Ці компоненти розташовані у вертикальному порядку, забезпечуючи зручний доступ до кожного елемента. Інтерфейс виглядає чистим і простим, полегшуючи процес аутентифікації користувачів. Усі елементи розміщені послідовно для легкого використання.

Сторінка відновлення паролю (рис. 4.3) створена для забезпечення зручного та швидкого відновлення доступу до облікового запису. Її дизайн простий та інтуїтивно зрозумілий, а функціонал спрямований на мінімізацію часу, потрібного для відновлення пароля.

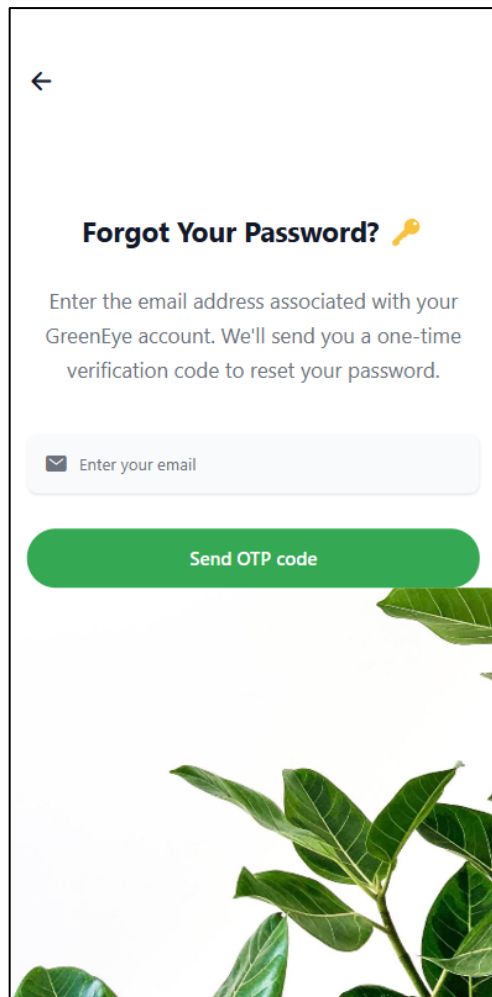


Рисунок 4.3 – Сторінка відновлення паролю

Екран для відновлення пароля облікового запису складається з декількох компонентів. Зверху розташований великий заголовок, а під заголовком

знаходиться інструкція, яка пояснює користувачам, що вони повинні ввести свою електронну адресу, пов'язану з акаунтом, щоб отримати одноразовий код для скидання пароля.

Далі йде текстове поле з попередньо введеною електронною адресою, що дає зрозуміти користувачам, де саме вводити свої дані. Під текстовим полем знаходиться кнопка, яка запускає процес відправлення коду для скидання пароля на вказану адресу. Усі компоненти екрану розміщені зручно і логічно, що полегшує процес відновлення доступу для користувачів. Інтерфейс має чистий та зрозумілий дизайн, що сприяє інтуїтивному користуванню.

На рис. 4.4 представлено сторінку для введення OTP (One-Time Password), який складається з декількох компонентів.

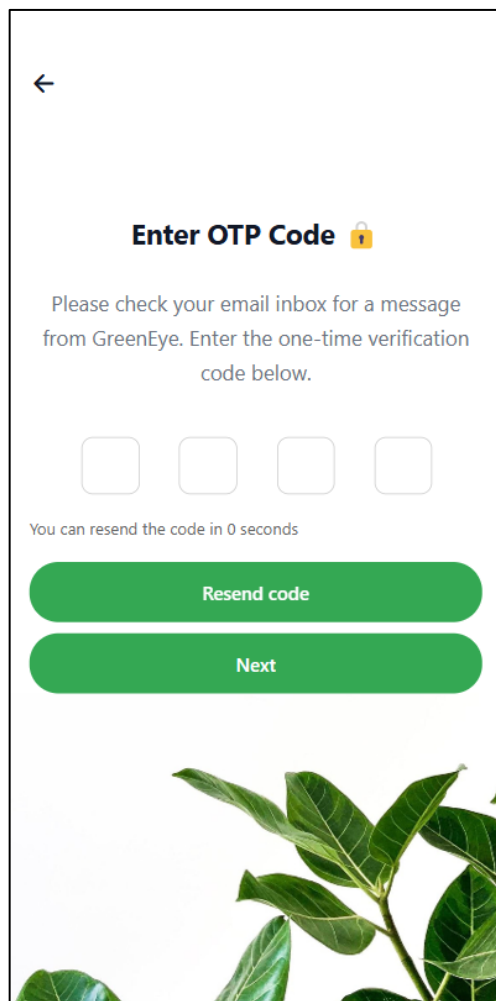


Рисунок 4.4 – Сторінка для введення OTP

На верху розташований заголовок, а під заголовком знаходиться інструкція, яка пояснює, що користувач повинен перевірити свою електронну пошту на наявність повідомлення і ввести код для підтвердження.

Нижче розташовані чотири порожні поля для введення OTP коду. Під цими полями є повідомлення, яке вказує, що користувач зможе повторно відправити код через 60 секунд.

У нижній частині екрану розташовані дві кнопки: для повторного відправлення коду та для переходу до наступного кроку.

Усі елементи екрану розміщені логічно та зрозуміло, що полегшує користувачам процес введення одноразового пароля. Інтерфейс має чіткий та інтуїтивно простий дизайн.

На рис. 4.5 зображено сторінку зміни паролю для облікового запису.

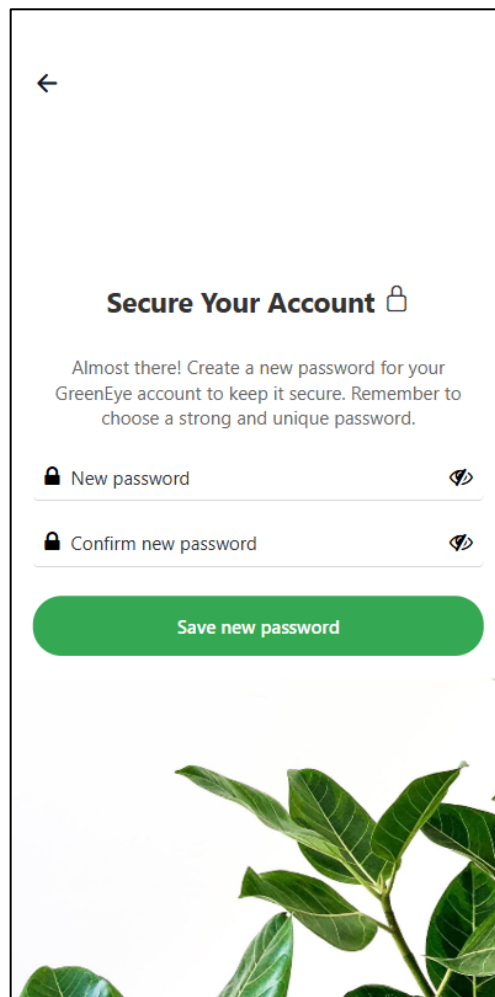


Рисунок 4.5 – Сторінка зміни паролю для облікового запису

У верхній частині сторінки розташований заголовок. Нижче розташовані два поля для введення нового паролю. Під полями для введення паролю є кнопка збереження нового паролю.

Усі елементи екрану розміщені логічно та зрозуміло, що полегшує користувачам процес зміни пароля. Інтерфейс має чіткий та інтуїтивно зрозумілий дизайн.

На рис. 4.6 зображено сторінку успішного підтвердження зміни паролю, яка складається з декількох компонентів.

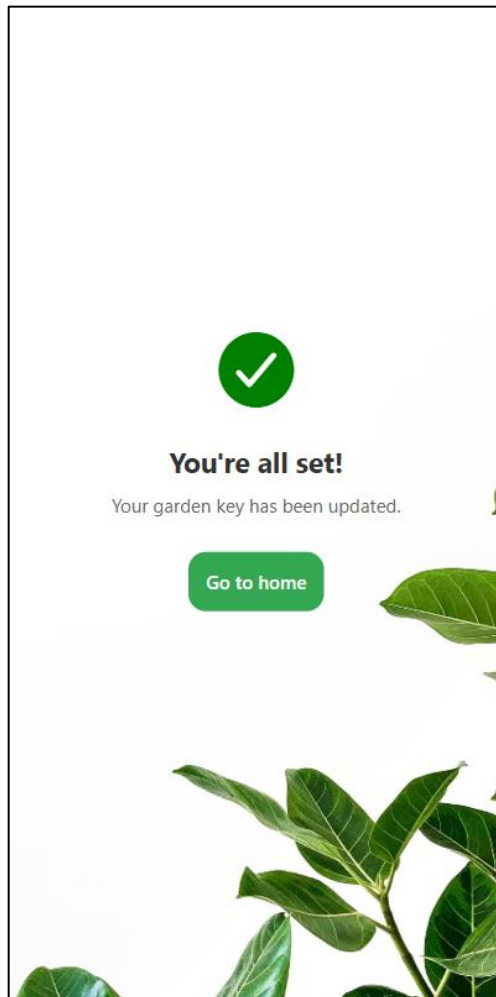


Рисунок 4.6 – Сторінка успішного підтвердження зміни паролю

У верхній частині виводиться повідомлення про успішне завершення процесу. Нижче знаходиться повідомлення підтвердження зміни паролю. На



нижній частині екрану знаходиться кнопка, яка повертає користувача на головну сторінку після оновлення пароля.

Інтерфейс виглядає інтуїтивно зрозумілим і легким для використання, що забезпечує позитивний досвід користувача. Усі елементи логічно розташовані для зручного користування.

На рис. 4.7 зображено головну сторінку застосунку, яка має кілька ключових розділів.

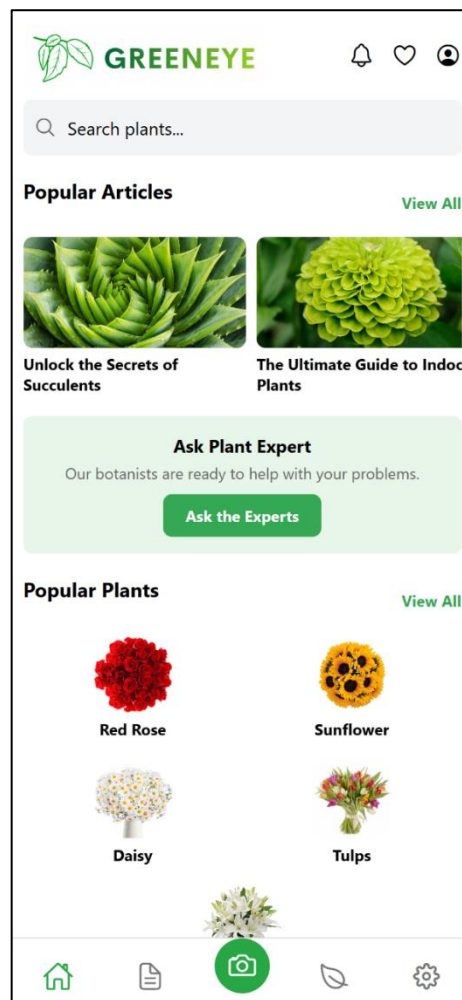


Рисунок 4.7 – Головну сторінка застосунку

У верхній частині екрану є поле для пошуку. Під полем пошуку розташовано розділ з популярними статтями.

Нижче знаходиться кнопка, яка надає доступ до консультацій у фахівців. Під кнопкою розташований розділ з популярними рослинами. У нижній частині екрану

Програмне забезпечення ідентифікації рослин на основі методів штучного інтелекту знаходиться панель навігації. Інтерфейс логічно організований, що забезпечує зручність користування.

На рис. 4.8 зображено сторінку з розділом про популярні статті, який присвячений догляду за рослинами.

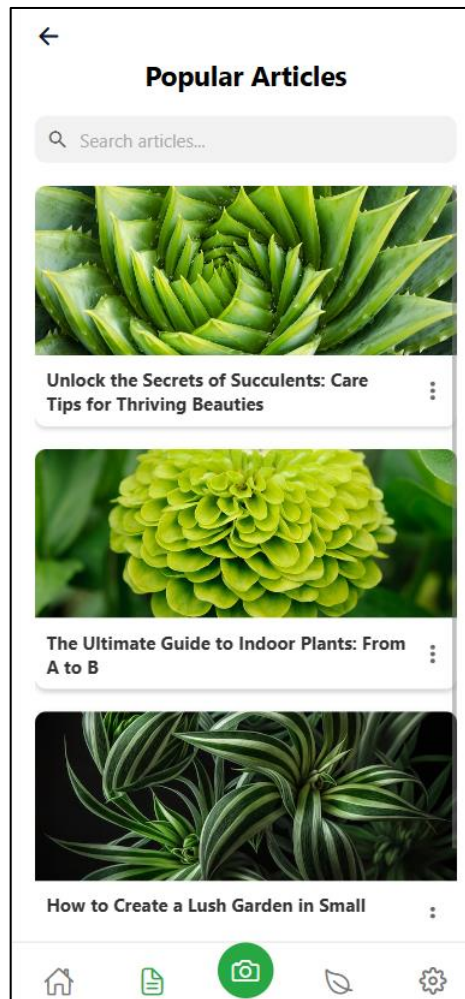


Рисунок 4.8 – Сторінка з розділом про популярні статті

У верхній частині екрану є поле для пошуку статей. Нижче розташовані статті з зображеннями рослин. Кожна стаття має кнопку додаткових опцій.

Інтерфейс інтуїтивно зрозумілий, що забезпечує зручність користування.

На рис. 4.9 зображено сторінку з розділом пошуку рослин.

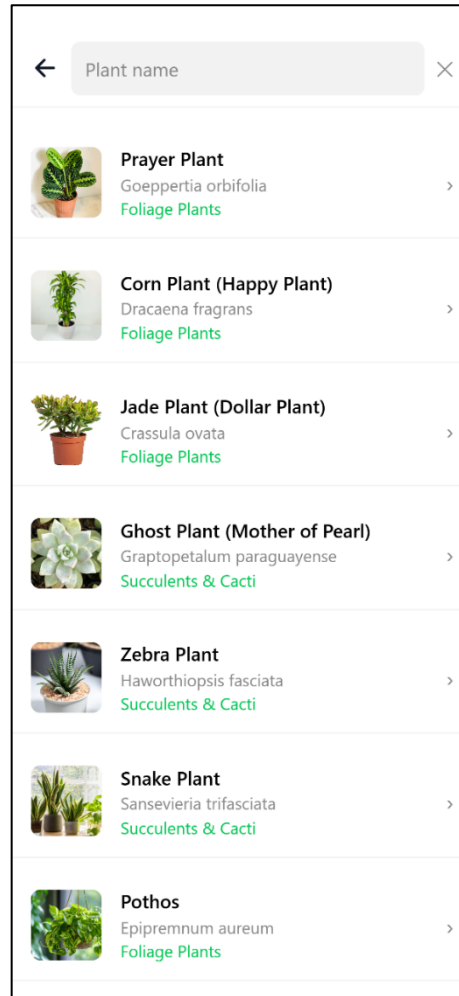


Рисунок 4.9 – Сторінка з розділом пошуку рослин

На сторінці пошуку рослин у верхній частині розташоване поле введення для пошуку рослин за назвою або видом. Нижче відображаються результати пошуку у вигляді карток, кожна з яких містить зображення рослини, її назву, вид і тип.

Інтерфейс інтуїтивно зрозумілий і забезпечує легкий доступ до інформації, дозволяючи швидко знаходити необхідну рослину. Кожна картка має кнопку додаткових опцій для перегляду детальнішої інформації або взаємодії з нею.

На рис. 4.10 зображено сторінку детальної інформації про рослину.

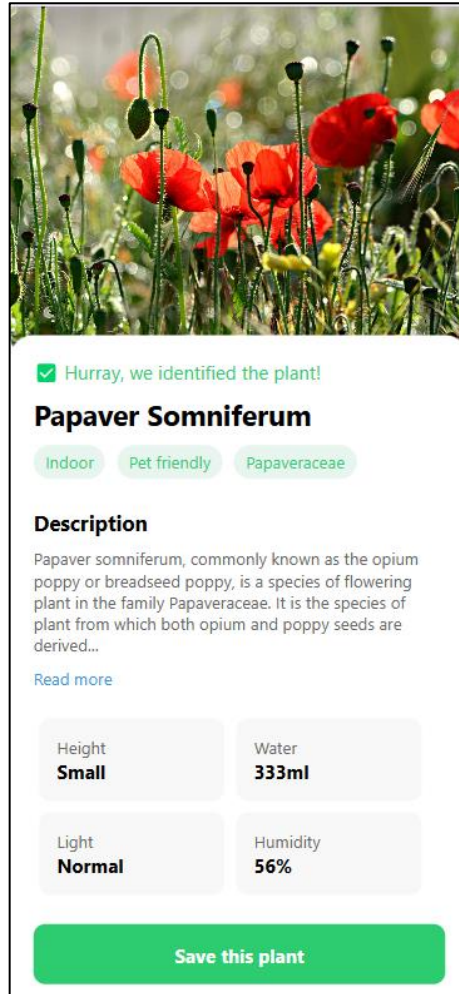


Рисунок 4.10 – Сторінка детальної інформації про рослину

Сторінка детальної інформації про рослину містить всю необхідну інформацію для ознайомлення з обраним екземпляром. У верхній частині розташоване фото рослини, під яким вказано її назву. Нижче представлено основні властивості, які притаманні цій рослині, включаючи її детальний опис.

Додатково відображається інформація про висоту рослини, вимоги до поливу, рівень освітлення та оптимальну вологість. У нижній частині сторінки знаходиться кнопка для збереження цієї рослини в розділ обраного, що дозволяє користувачу швидко знайти її в майбутньому. Інтерфейс структурований і зручний для користування.

На рис. 4.11 зображено сторінку налаштувань.

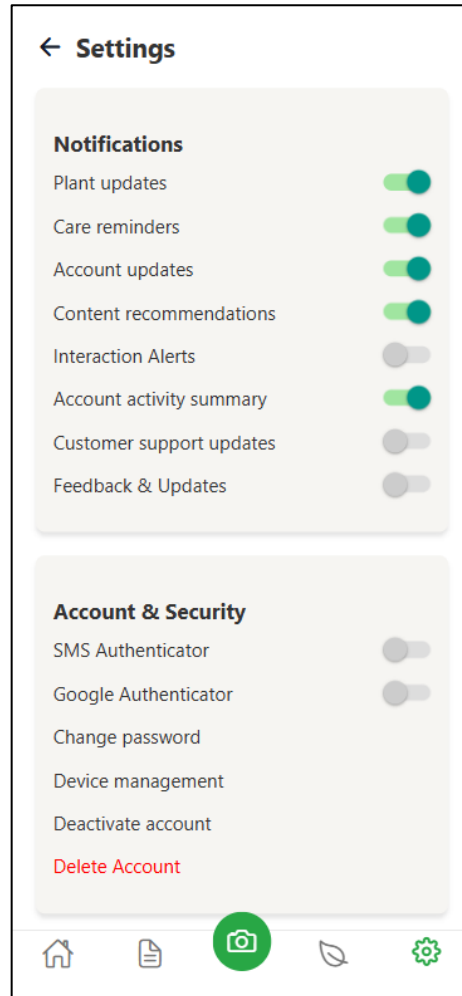


Рисунок 4.11 – Сторінка налаштувань

Сторінка налаштувань складається з кількох секцій, що дозволяють користувачеві легко налаштувати застосунок.

- *Notifications (Сповіщення)*. Тут користувач може налаштувати типи сповіщень, які хоче отримувати, та вибрати спосіб їх надходження.
- *Account & Security (Акаунт та безпека)*. Ця секція дозволяє змінити пароль, налаштувати двофакторну автентифікацію, а також переглянути інформацію про доступні пристрої.
- *Privacy Settings (Налаштування конфіденційності)*. Тут користувач може налаштувати, хто може бачити його інформацію та керувати доступом до особистих даних.

– *App Preferences (Переваги застосунка)*. В цій секції можна налаштувати загальні параметри використання застосунка, такі як темна/світла тема, а також інші персоналізовані налаштування.

Інтерфейс сторінки налаштувань інтуїтивно зрозумілий і дозволяє користувачеві швидко змінювати необхідні опції для покращення досвіду використання застосунка.

На рис. 4.12 зображено сторінку повідомлень (нотифікацій).

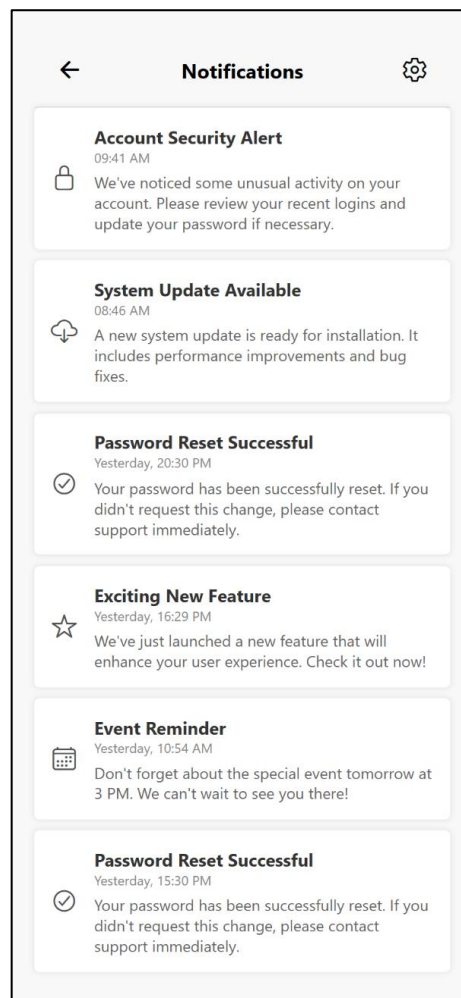
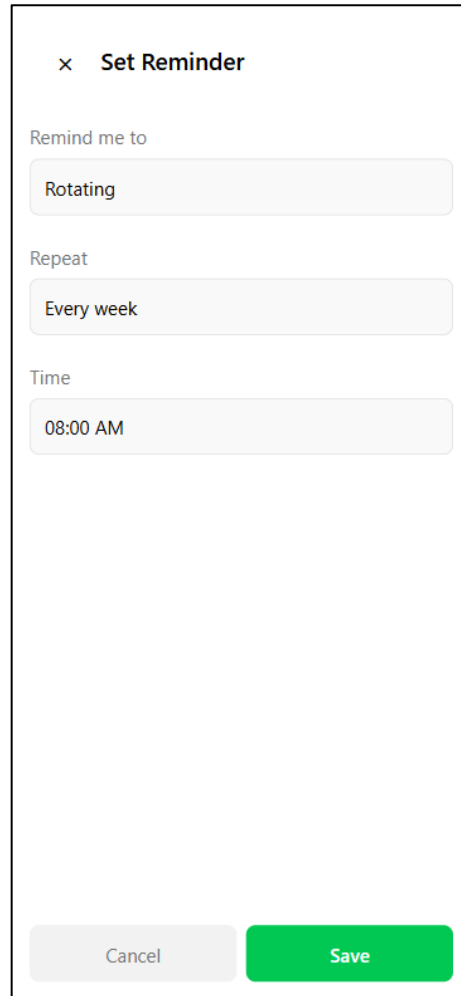


Рисунок 4.12 – Сторінка повідомлень (нотифікацій)

Сторінка повідомлень (нотифікацій) відображає важливі сповіщення для користувача щодо діяльності в застосунку. Кожне повідомлення супроводжується відповідною іконкою для швидкої ідентифікації типу повідомлення.



The image shows a mobile application dialog box titled "Set Reminder". It has a close button (X) in the top left corner. The dialog contains three input fields: "Remind me to" with the value "Rotating", "Repeat" with the value "Every week", and "Time" with the value "08:00 AM". At the bottom, there are two buttons: "Cancel" (grey) and "Save" (green).

Рисунок 4.13 – Сторінка встановлення нагадувань

Сторінка встановлення нагадувань (рис. 4.13) дозволяє користувачеві налаштувати персоналізовані нагадування для догляду за рослинами. На сторінці є кілька полів для вибору параметрів нагадувань:

- “*Remind me to*”. Це поле дозволяє вибрати тип догляду, для якого необхідно встановити нагадування, наприклад, полив, обертання рослини, підрізання тощо.

- “*Repeat*”. У цьому полі користувач може вибрати, як часто він хоче отримувати нагадування для обраної дії. Доступні опції включають щодня, щотижня, щомісяця тощо.

- “*Time*”. Тут користувач може вибрати точний час, коли він хоче отримувати нагадування (наприклад, щодня о 9:00 або кожен понеділок о 15:00).

Інтерфейс сторінки простий і зручний для налаштування нагадувань, дозволяючи користувачеві легко налаштувати періодичні нагадування для догляду за рослинами.

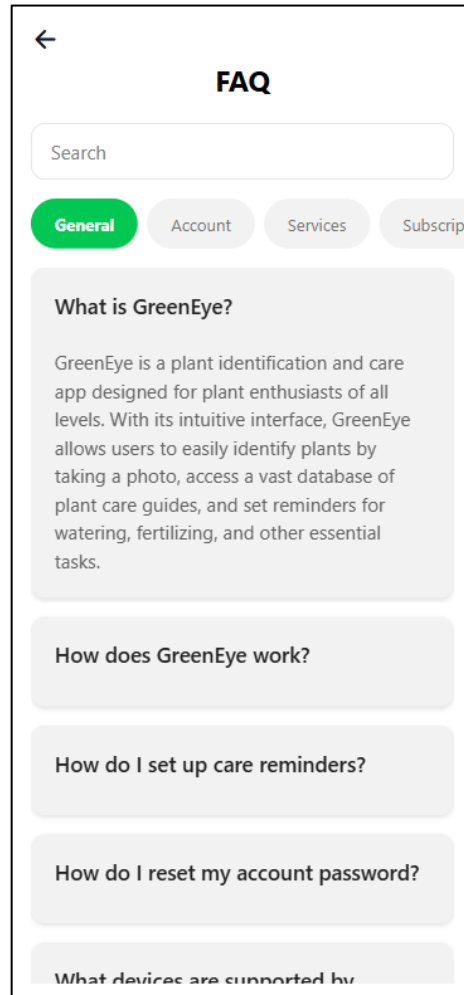


Рисунок 4.14 – Сторінка FAQ

Сторінка FAQ (рис. 4.14) містить список найбільш поширених запитань користувачів і відповідей на них. Вона розроблена для того, щоб допомогти користувачам швидко знаходити інформацію та вирішувати питання без необхідності звертатися до служби підтримки.

На сторінці є структуровані категорії запитань, що дозволяють швидко знайти відповідь на конкретну тему. Кожне питання супроводжується чіткою та зрозумілою відповіддю. Для зручності користувачів питання можна розгортати або згорнути, що дозволяє не перенавантажувати сторінку зайвою інформацією.



Інтерфейс сторінки зручний для навігації і дозволяє швидко знайти відповідь на питання, з яким може зіткнутися користувач.

На рис. 4.15 зображено сторінку пошуку в застосунку, яка демонструє відсутність результатів за введеним запитом.

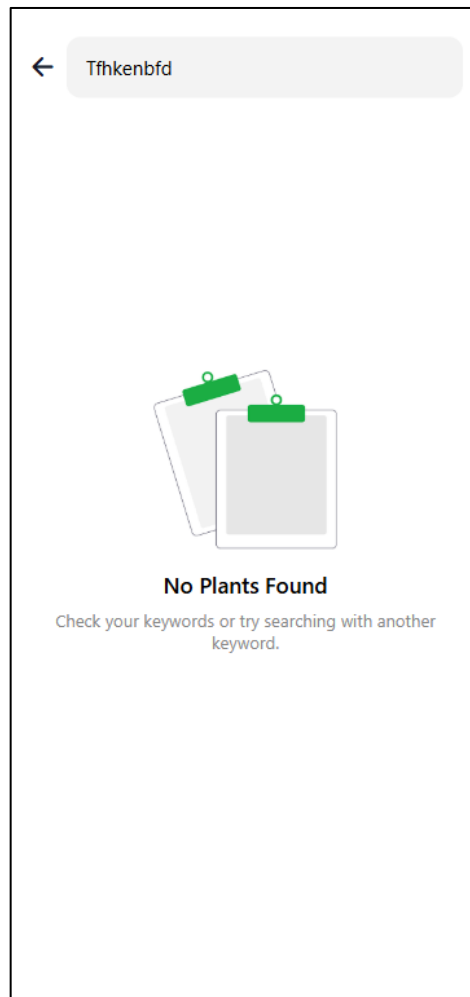


Рисунок 4.15 – Сторінка пошуку

У верхній частині екрану знаходиться текстове поле для пошуку, поруч із кнопкою повернення назад, позначеною іконкою стрілки. Поле пошуку містить текст, введений користувачем.

У центральній частині сторінки розташований значок, який символізує відсутність результатів — два порожніх блоки зі скріпками. Нижче зображення знаходиться текст із повідомленням “*No Plants Found*”, який інформує користувача про те, що за введеним запитом немає відповідних рослин у базі даних.

Під цим повідомленням наведено підказку для спроби пошуку за іншим запитом.

Інтерфейс має мінімалістичний і зручний дизайн, орієнтований на швидке взаємодію користувача з пошуковою системою застосунка.

Сторінка пошуку рослин включає розділи для зручної навігації та пошуку (рис. 4.16).

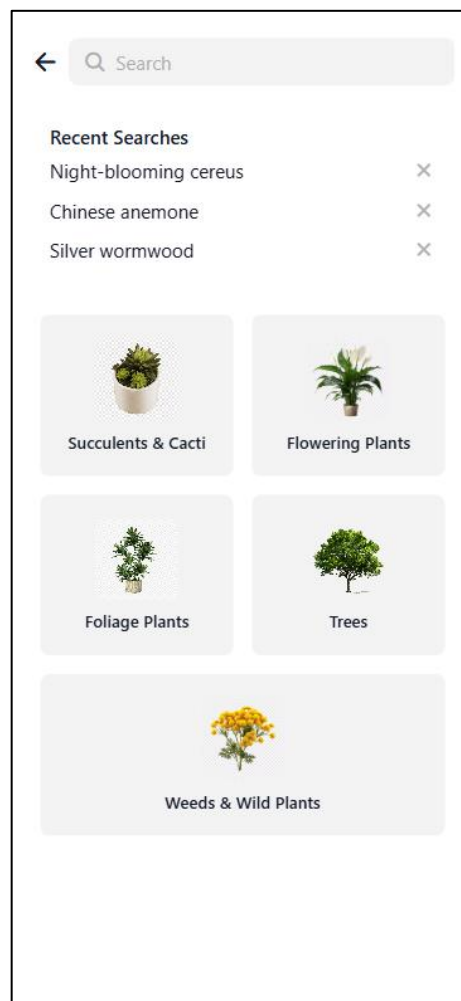


Рисунок 4.16 – Сторінка пошуку рослин

У верхній частині розташований рядок пошуку, під яким знаходиться розділ останніх запитів. Тут користувач може побачити свої останні пошуки, із можливістю видалення кожного пункту через кнопку.

Інтерфейс візуально структурований та інтуїтивний, що дозволяє легко знаходити потрібну інформацію.

## **Висновки до розділу 4**

У даному розділі здійснено ґрунтовний аналіз та реалізацію ключових аспектів розробки системи, що базується на сучасних методах штучного інтелекту для ідентифікації рослин. Описано процес вибору програмних інструментів, технологічного стеку, а також основні етапи створення функціональних модулів системи.

На основі проведеного аналізу було обґрунтовано доцільність використання сучасних мов програмування, бібліотек та баз даних, які забезпечують високу продуктивність та гнучкість системи. Особливу увагу приділено використанню методів машинного навчання для створення модулів ідентифікації рослин, які є центральною частиною застосунку. Також детально описано особливості реалізації функцій управління даними, забезпечення взаємодії між користувачами та адміністраторами, а також підтримки багатокористувацького доступу до системи.

Розробка застосунку орієнтована на забезпечення доступності для користувачів різних категорій, таких як науковці або звичайні користувачі, за допомогою інтуїтивно зрозумілого інтерфейсу та оптимізованої архітектури. Завдяки ретельному проектуванню та реалізації, система здатна обробляти великі обсяги інформації, забезпечуючи її швидкий доступ та ефективну обробку.

Таким чином, програмне забезпечення “GreenEye” створено відповідно до сучасних стандартів програмної інженерії, що забезпечує його надійність, масштабованість та ефективність. Реалізовані функціональні можливості повністю відповідають вимогам, встановленим на етапі проектування, і демонструють потенціал для подальшого розвитку та інтеграції нових технологій.

## ВИСНОВКИ

У ході виконання кваліфікаційної магістерської роботи підвищено ефективність і точність розпізнавання рослин за рахунок створення універсальної системи, призначеної для ідентифікації рослин на основі методів штучного інтелекту.

Для досягнення вказаної мети вирішено поставлені завдання:

1. проведено аналіз існуючих рішень та аналогічних програмних забезпечень;
2. розроблено технічне завдання на створення системи;
3. розроблено дизайн та інтерфейс користувацької частини застосунка;
4. реалізовано функціональні можливості застосунку, включаючи модуль ідентифікації рослин;
5. створено базу даних для збереження інформації про рослини;
6. забезпечено механізм автоматичної обробки зображень рослин за допомогою нейронних мереж;
7. реалізовано функції пошуку та надання додаткової інформації про ідентифіковані рослини;
8. протестовано застосунок та забезпечено його стабільну роботу.

Розглянуто етапи реалізації проекту, розроблено UML-діаграми та зроблено огляд задіяних технологій для розробки. Описано дизайн системи та проведено тестування задля коректної роботи системи. Результатом проведеної роботи є система ідентифікації рослин на основі ШІ.

### ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. PlantNet: Plant Identification. URL: <https://apps.apple.com/ru/app/plantnet/id600547573> (Last accessed: 10.09.2024).
2. iNaturalist. URL: <https://apps.apple.com/ua/app/inaturalist/id421397028?l=uk> (Last accessed: 12.09.2024).
3. PictureThis: Flowers and Trees. URL: <https://apps.apple.com/ru/app/picturethis> (Last accessed: 15.09.2024).
4. A. Begue, V. Kowlessur, U. Singh, F. Mahomoodally, and S. Pudaruth, “Automatic recognition of medicinal plants using machine learning techniques,” *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 4, 2017.
5. M. M. Amlekar and A. T. Gaikwad, “Plant classification using image processing and neural network,” in *Data Management, Analytics and Innovation*, V. E. Balas, N. Sharma, and A. Chakrabarti, Eds. Singapore: Springer Singapore, 2019, pp. 375–384.
6. J. Carranza-Rojas, H. Goeau, P. Bonnet, E. Mata-Montero, and A. Joly, “Going deeper in the automated identification of herbarium specimens,” *BMC Evolutionary Biology*, vol. 17, 2017.
7. H. Zhu, Q. Liu, Y. Qi, X. Huang, F. Jiang, and S. Zhang, “Plant identification based on very deep convolutional neural networks,” *Multimedia Tools and Applications*, vol. 77, 2018.
8. K. P. Ferentinos, “Deep learning models for plant disease detection and diagnosis,” *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018.
9. M. Brahim, M. Arsenovic, S. Laraba, S. Sladojevic, K. Boukhalfa, and A. Moussaoui, *Deep Learning for Plant Diseases: Detection and Saliency Map Visualisation*. Cham: Springer International Publishing, 2018, pp. 93–117.

10. C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, jul 2019.
11. H. Yalcin and S. Razavi, "Plant classification using convolutional neural networks," 2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics), Tianjin, pp. 1-5, 2016.
12. Choi Sungbin. "Plant Identification with Deep Convolutional Neural Network: SNUMedinfo at LifeCLEF Plant Identification Task 2015." In CLEF (Working Notes). 2015.
13. Xiao, Qingguo, Guangyao Li, Li Xie, and Qiaochuan Chen. "Real-world plant species identification based on deep convolutional neural networks and visual attention." *Ecological Informatics* 48 (2018): 117-124.
14. Secure and resizable cloud compute – Amazon EC2 – Amazon Web Services. URL: <https://aws.amazon.com/ec2/> (Last accessed: 15.10.2024).
15. Fully Managed Relational Database – Amazon RDS – Amazon Web Services. URL: <https://aws.amazon.com/rds/> (Last accessed: 18.10.2024).
16. Cloud Object Storage – Amazon S3 – Amazon Web Services. URL: <https://aws.amazon.com/s3/> (Last accessed: 20.10.2024).
17. Understanding GitHub Actions – GitHub Docs. URL: <https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions> (Last accessed: 15.11.2024).
18. Як будувати UML-діаграми. Розбираємо три найпопулярніші варіанти. URL: <https://dou.ua/forums/topic/40575/> (Дата звернення: 18.11.2024).
19. Елементи UML. Діаграма випадків використання. URL: <https://docs.kde.org/trunk5/uk/umbrello/umbrello/uml-elements.html> (Дата звернення: 20.11.2024).

## ДОДАТОК А Апробація результатів КМР

# Апробація результатів роботи

Апробація результатів кваліфікаційної магістерської роботи відбулася під час:

**XXVII Всеукраїнської науково-практичної конференції «Могилянські читання», секції «Технічні науки», підсекції «Моделі, методи та засоби програмної інженерії» (Миколаїв, 06–10 листопада 2024 р.).**

**Гайсюк А. В., Решетник Ю. В.** Система ідентифікації рослин на основі методів штучного інтелекту .....42



Додаток А.1 – Апробація результатів КМР